innovar healthcare

Innovar Channel History Plugin

Git v2.1

Created: Sep 16, 2025
Updated: Sep 16, 2025

innovar healthcare

Table of Contents

innovar healthcare

Introduction

Welcome to the Channel History plugin documentation for BridgeLink. This plugin is designed to allow you to utilize Git version control automatically with BridgeLink interface development.
NOTE: Channel History Plugin can be used with both BridgeLink and Mirth Connect.

Why channel history?

During the development of healthcare interfaces, the importance of version control can not be overemphasized. As more and more clients are on board with your team, it is challenging to track the progress of programming development on the BridgeLink engine. Therefore, Innovar built this channel history plugin that not only provides the differing view of commits within the BridgeLink administrator tool but automatically pushes the commits to your remote repository. With this powerful plugin, you will no longer lose track of your code and will be able to restore to the previous version quickly.

Key Features

- History and Rollback: The ability to view the history of changes and roll back to previous versions is crucial. If a new change introduces a bug or an issue, it can be reverted to a stable state quickly.
- Remote Repository: Storing BridgeLink channels and code templates on GitHub provides a remote backup. This is essential for disaster recovery, ensuring that configurations are not lost due to local hardware failures or other issues.

Getting Started

Before you dive into the documentation, make sure to review the installation prerequisites and check compatibility with your existing BridgeLink setup. The subsequent sections will guide you through the configuration and how to use the Channel History plugin.

Let's get started!

Installation

If you are subscribed to the BridgeLink package from Innovar Healthcare on the AWS Marketplace, the extension should be pre-installed in the ‘BridgeLink Standard Edition an Open Source Mirth Connect Fork’, ‘BridgeLink Enterprise Edition an Open Source Mirth Connect Fork versions’.

If, for some reason, you need to reinstall or update the plugin, you can do this from the BridgeLink application.

You can find the latest version of the plugin, as well as older releases and user guides, on our Innovar Website or here:
https://innovar-userdocuments.s3.us-east-2.amazonaws.com/index.html

1. While logged into BridgeLink, click on **Extensions**, then at the bottom of the screen, click **Browse**.
2. A new window will pop up to allow you to browse for your plugin zip file on your local machine.
3. Browse to the appropriate zip file and click **Open**. Once back on the Extensions screen, your file path should be filled in.
4. Click **Install** to upload the file.
5. Once completed, you will need to restart the BridgeLinkt Service on the remote server.

Prerequisites

You will need to create a new repository on a version control platform (such as GitHub, Bitbucket, AWS CodeCommit, etc.) and attach your SSH public key to the remote repository. Below we will provide information on Bitbucket and Github.

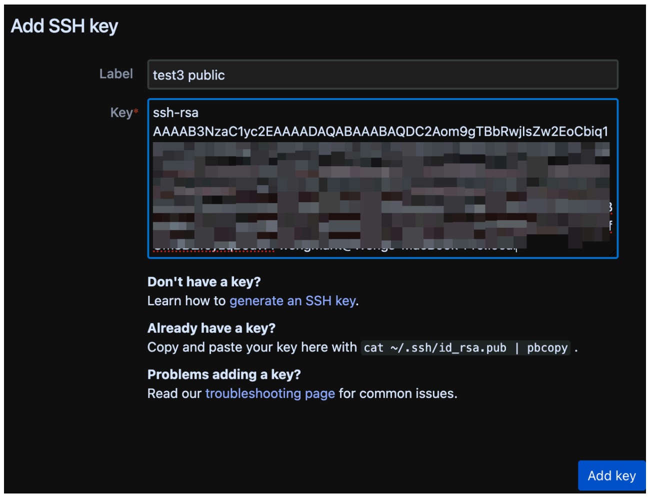
Steps to Configure Your SSH Key in Bitbucket

1. Use the following command to generate an SSH key pair. It will create a key pair named "test3", where "test3" is the private key and "test3.pub" is the public key.

```
ssh-keygen -t rsa -b 2048 -f test3
```

2. Go to your version control tool and add the public key to your user account or target repository.

Example for Bitbucket:



Add SSH key

Label: test3 public

Key*: ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDC2Aom9gTBbRwjlsZw2EoCbiq1 [redacted]

Don't have a key?
Learn how to generate an SSH key.

Already have a key?
Copy and paste your key here with `cat ~/.ssh/id_rsa.pub | pbcopy` .

Problems adding a key?
Read our troubleshooting page for common issues.

3. Please run the following command to convert your ssh private key into PEM format. Make sure you back up your private key first.

```
cp test3 test3_backup
ssh-keygen -p -f test3 -m PEM
```

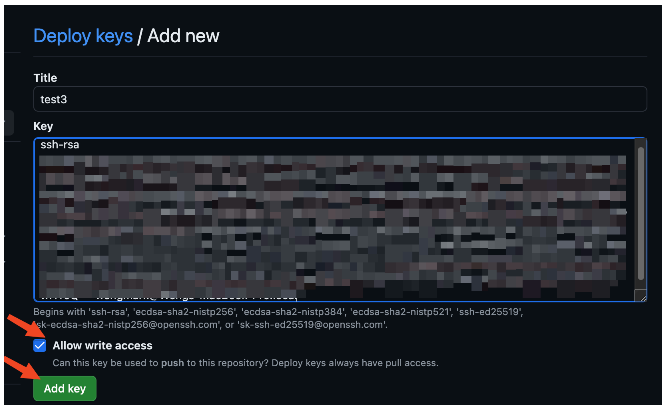
Then you are good to upload the private key PEM file to the channel history plugin setting!

Configure SSH key on GitHub

1. Please use the following command to generate the ssh key:It will create a key pair “test3”. The “test3” file is the private key and the “test3.pub” file is the public key.

```
ssh-keygen -t rsa -b 4096 -E sha256 -m PEM -f test3
```

2. Go to your GitHub repository. Select “Settings” > “Deploy keys” > “Add deploy keys”. Paste your ssh public key and click “Add key”.


Deploy keys / Add new

Title

test3

Key

ssh-rsa
[redacted]

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'.

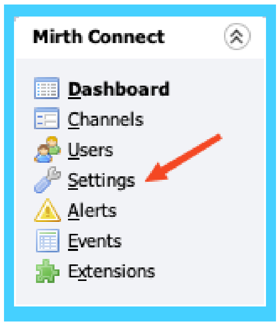
[✓] Allow write access

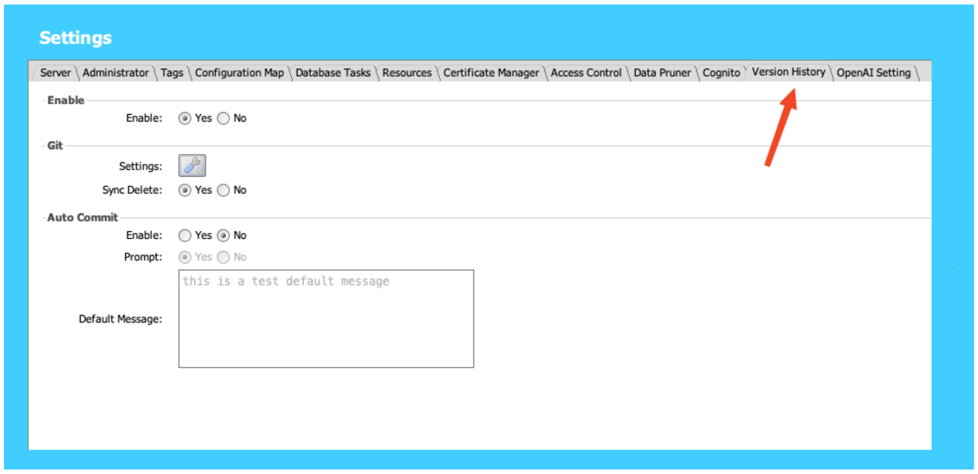
Can this key be used to **push** to this repository? Deploy keys always have pull access.

Then you are good to upload the private key PEM file to the channel history plugin setting!

Plugin Configuration

1. In the BridgeLink window, click on **Settings**. You will see a new tab for **Version History**.





Enable

Enable: [✓ Yes] No

Git

Settings:

Sync Delete: [✓ Yes] No

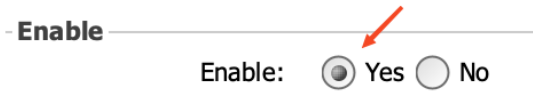
Auto Commit

Enable: Yes [✓ No]

Prompt: [✓ Yes] No

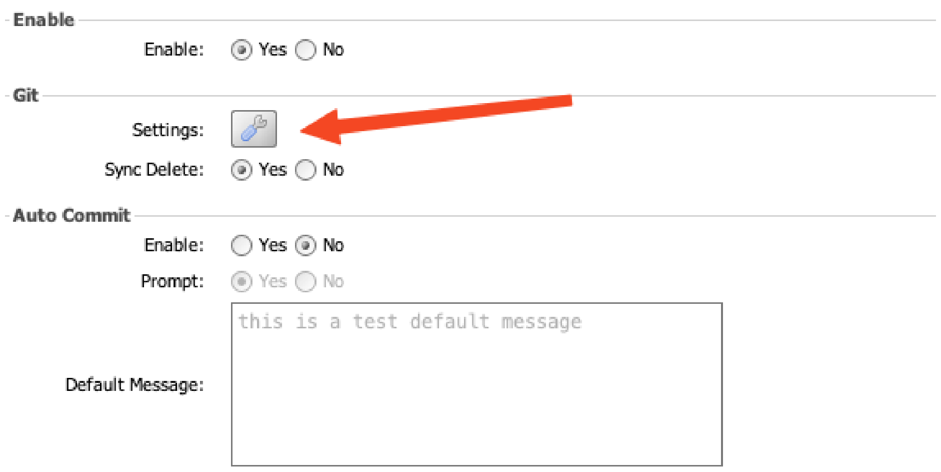
Default Message: this is a test default message

2. Please check **Enable** to enable the plugin.


Enable

Enable: [✓ Yes] No

3. Under the Git field click the **Settings tool** and a pop up window will appear.


Enable

Enable: [✓ Yes] No

Git

Settings:

Sync Delete: [✓ Yes] No

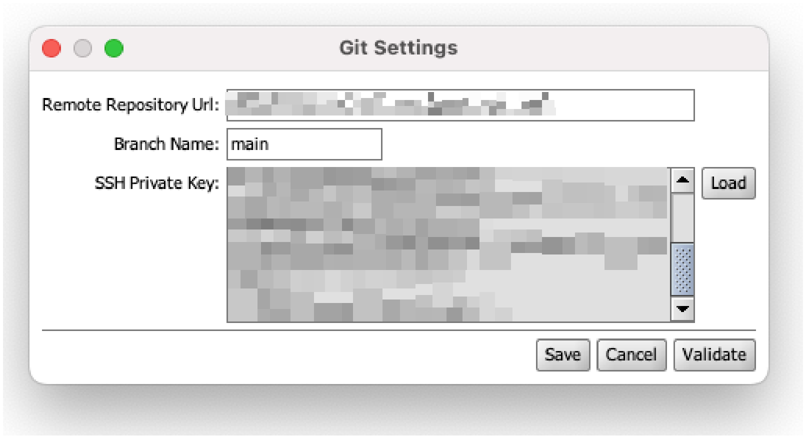
Auto Commit

Enable: Yes [✓ No]

Prompt: [✓ Yes] No

Default Message: this is a test default message

4. On the Git Settings window, Add your remote repository URL, your branch name and your SSH Private Key. For the Private Key, you can paste it on the text area or simply click **Upload** and select the key file from your directory.

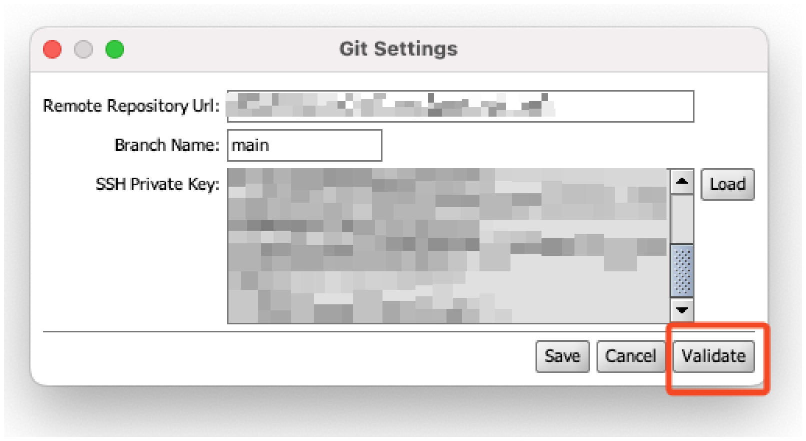

Git Settings

Remote Repository Url:

Branch Name: main

SSH Private Key:

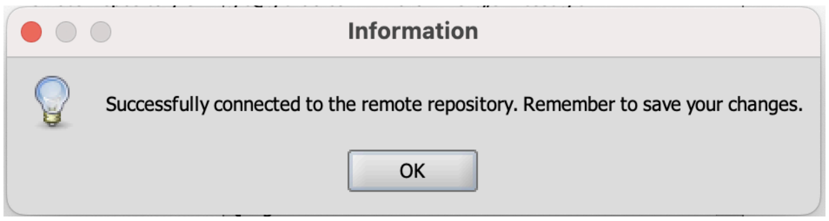
5. After adding the information, click **Validate** to make sure it works and you will see a successful message. Make sure you click **Save** after validating your Settings.


Git Settings

Remote Repository Url: [redacted]

Branch Name: main

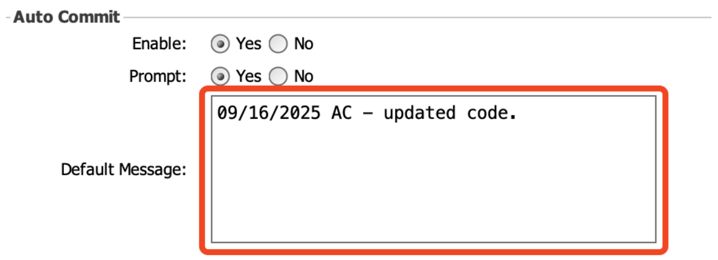
SSH Private Key: [redacted]


Information

Successfully connected to the remote repository. Remember to save your changes.

6. Under the Auto Commit field, you can Enable it to allow the application to automatically commit changes to the remote repository. If you prefer to commit manually, simply select **No**.

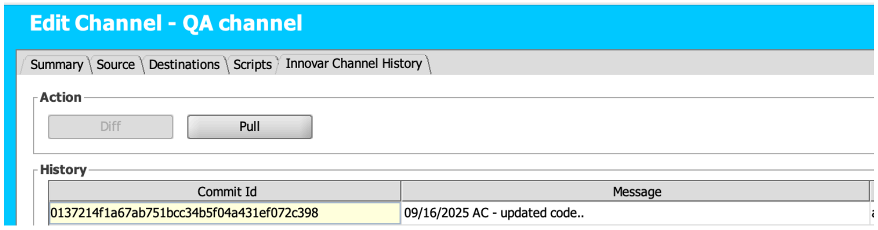
7. You can also add a default message that will appear when the code Auto Commits.


Auto Commit

Enable: [✓ Yes] No

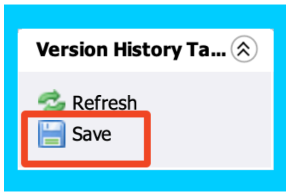
Prompt: [✓ Yes] No

Default Message: 09/16/2025 AC – updated code.


Edit Channel - QA channel

Commit Id	Message
0137214f1a67ab751bcc34b5f04a431ef072c398	09/16/2025 AC - updated code..

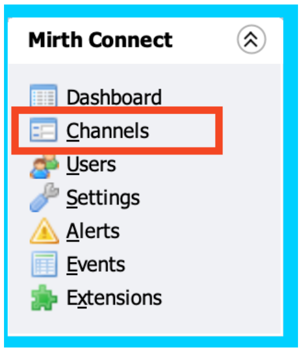
8. Make sure you click **Save** after making your changes on the Task Panel. The channel history plugin will clone your remote repo to /opt/mirthconnect/appdata/InnovarHealthcare-version-control folder

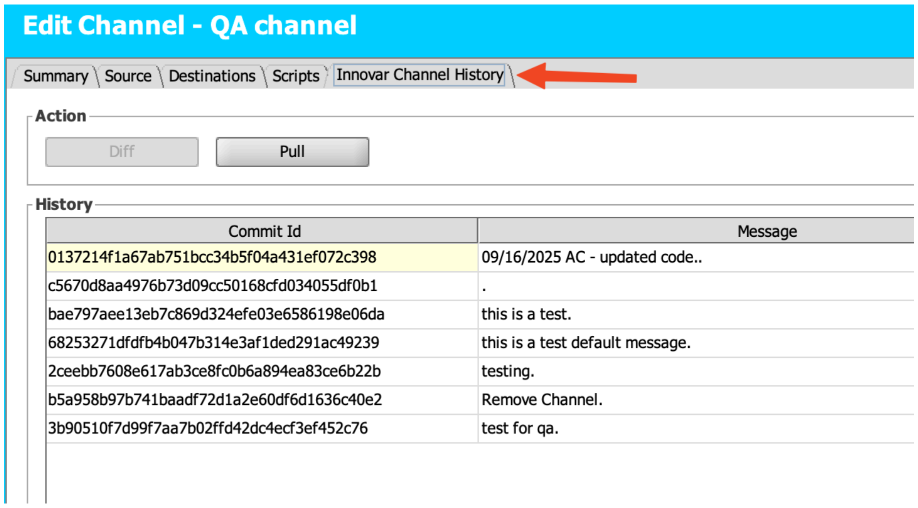


View Channel Code History

After configuring the settings, you can view the code history for channels. To do so, make a change to the target channel and save it.

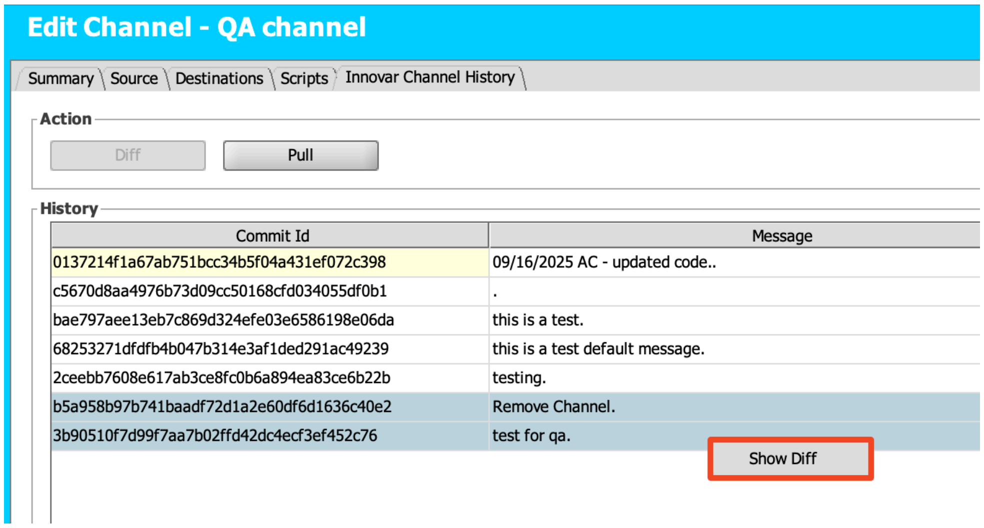
1. Please go to **Channels** and select the target channel. Go to the **Innovar Channel History** tab.



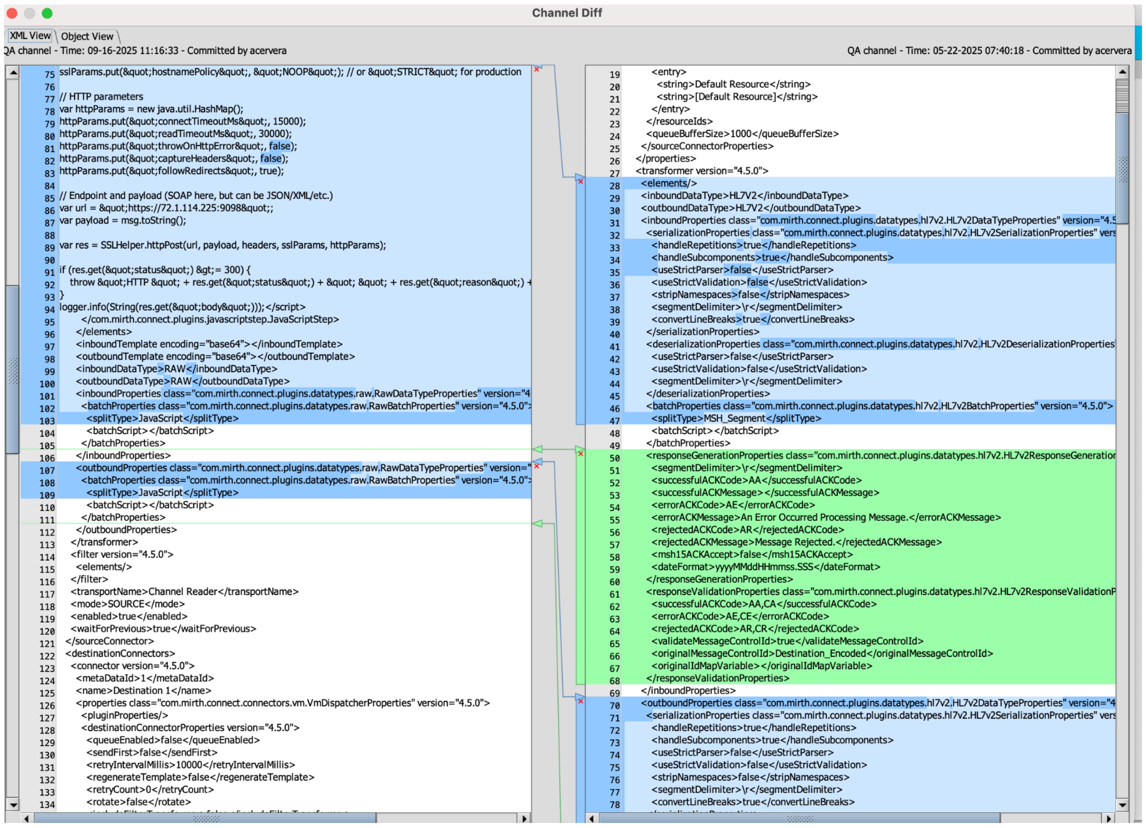


Commit Id	Message
0137214f1a67ab751bcc34b5f04a431ef072c398	09/16/2025 AC - updated code..
c5670d8aa4976b73d09cc50168cfd034055df0b1	.
bae797aee13eb7c869d324efe03e6586198e06da	this is a test.
68253271dfdfb4b047b314e3af1ded291ac49239	this is a test default message.
2ceebb7608e617ab3ce8fc0b6a894ea83ce6b22b	testing.
b5a958b97b741baadf72d1a2e60df6d1636c40e2	Remove Channel.
3b90510f7d99f7aa7b02ffd42dc4ecf3ef452c76	test for qa.

2. Press the Control key on your keyboard, click on two commits that you want to compare and then click **Show Diff**. The Diff window shows the difference between 2 commits.



Commit Id	Message
0137214f1a67ab751bcc34b5f04a431ef072c398	09/16/2025 AC - updated code..
c5670d8aa4976b73d09cc50168cfd034055df0b1	.
bae797aee13eb7c869d324efe03e6586198e06da	this is a test.
68253271dfdfb4b047b314e3af1ded291ac49239	this is a test default message.
2ceebb7608e617ab3ce8fc0b6a894ea83ce6b22b	testing.
b5a958b97b741baadf72d1a2e60df6d1636c40e2	Remove Channel.
3b90510f7d99f7aa7b02ffd42dc4ecf3ef452c76	test for qa.



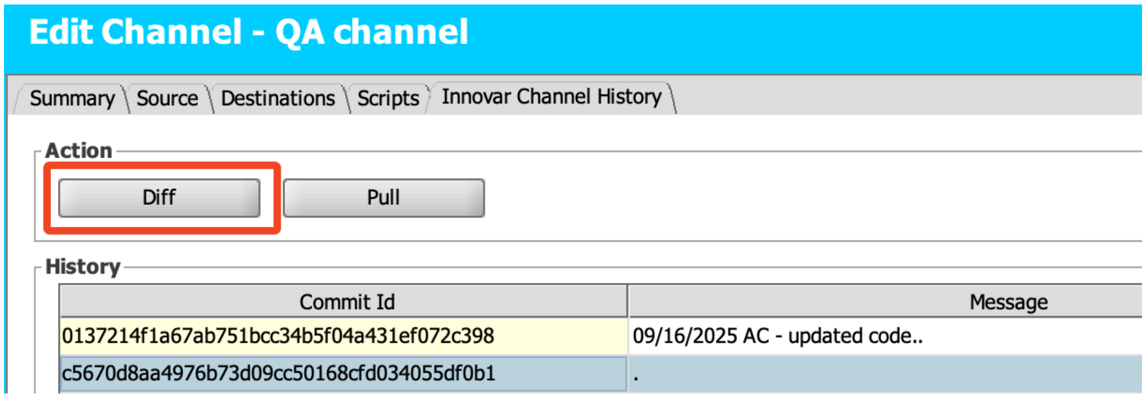
QA channel - Time: 09-16-2025 11:16:33 - Committed by acervera

```
sslParams.put(&quot;hostnamePolicy&quot;, &quot;NOOP&quot;); // or &quot;STRICT&quot; for production

// HTTP parameters
var httpParams = new java.util.HashMap();
httpParams.put(&quot;connectTimeoutMs&quot;, 15000);
httpParams.put(&quot;readTimeoutMs&quot;, 30000);
httpParams.put(&quot;throwOnHttpError&quot;, false);
httpParams.put(&quot;captureHeaders&quot;, false);
httpParams.put(&quot;followRedirects&quot;, true);

// Endpoint and payload (SOAP here, but can be JSON/XML/etc.)
var url = &quot;https://72.1.114.225:9098&quot;;
var payload = msg.toString();

var res = SSLHelper.httpPost(url, payload, headers, sslParams, httpParams);

if (res.get(&quot;status&quot;) &gt;= 300) {
  throw &quot;HTTP &quot; + res.get(&quot;status&quot;) + &quot; &quot; + res.get(&quot;reason&quot;) +
}
logger.info(String(res.get(&quot;body&quot;)));</script>
      </com.mirth.connect.plugins.javascriptstep.JavaScriptStep>
    </elements>
    <inboundTemplate encoding="base64"></inboundTemplate>
    <outboundTemplate encoding="base64"></outboundTemplate>
    <inboundDataType>RAW</inboundDataType>
    <outboundDataType>RAW</outboundDataType>
    <inboundProperties class="com.mirth.connect.plugins.datatypes.raw.RawDataTypeProperties" version="4
      <batchProperties class="com.mirth.connect.plugins.datatypes.raw.RawBatchProperties" version="4.5.0">
        <splitType>JavaScript</splitType>
        <batchScript></batchScript>
      </batchProperties>
    </inboundProperties>
    <outboundProperties class="com.mirth.connect.plugins.datatypes.raw.RawDataTypeProperties" version="
      <batchProperties class="com.mirth.connect.plugins.datatypes.raw.RawBatchProperties" version="4.5.0">
        <splitType>JavaScript</splitType>
        <batchScript></batchScript>
      </batchProperties>
    </outboundProperties>
  </transformer>
  <filter version="4.5.0">
    <elements/>
  </filter>
  <transportName>Channel Reader</transportName>
  <mode>SOURCE</mode>
  <enabled>true</enabled>
  <waitForPrevious>true</waitForPrevious>
</sourceConnector>
<destinationConnectors>
  <connector version="4.5.0">
    <metaDataId>1</metaDataId>
    <name>Destination 1</name>
    <properties class="com.mirth.connect.connectors.vm.VmDispatcherProperties" version="4.5.0">
      <pluginProperties/>
      <destinationConnectorProperties version="4.5.0">
        <queueEnabled>false</queueEnabled>
        <sendFirst>false</sendFirst>
        <retryIntervalMillis>10000</retryIntervalMillis>
        <regenerateTemplate>false</regenerateTemplate>
        <retryCount>0</retryCount>
        <rotate>false</rotate>
```

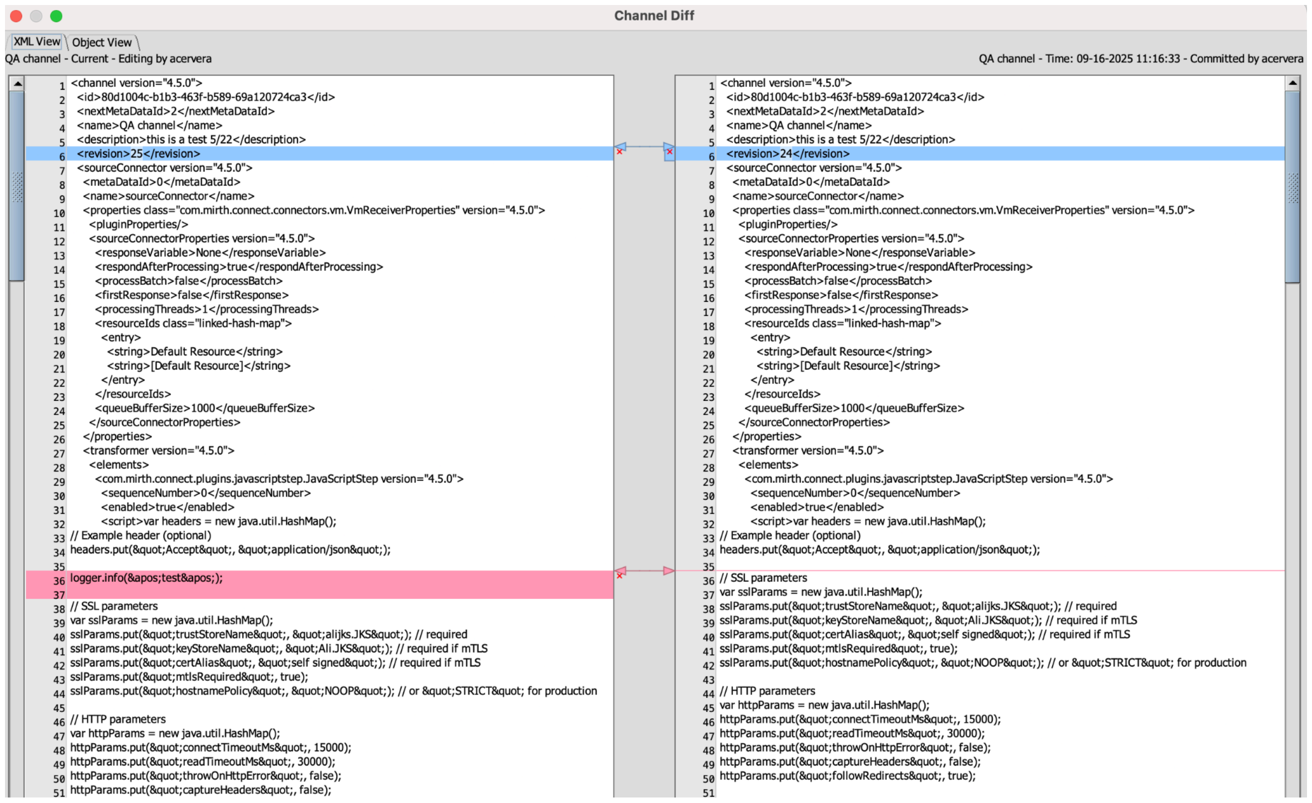
QA channel - Time: 05-22-2025 07:40:18 - Committed by acervera

```
        <entry>
          <string>Default Resource</string>
          <string>[Default Resource]</string>
        </entry>
      </resourceIds>
      <queueBufferSize>1000</queueBufferSize>
    </sourceConnectorProperties>
  </properties>
  <transformer version="4.5.0">
    <elements/>
    <inboundDataType>HL7V2</inboundDataType>
    <outboundDataType>HL7V2</outboundDataType>
    <inboundProperties class="com.mirth.connect.plugins.datatypes.hl7v2.HL7v2DataTypeProperties" version="4.5
      <serializationProperties class="com.mirth.connect.plugins.datatypes.hl7v2.HL7v2SerializationProperties" vers
        <handleRepetitions>true</handleRepetitions>
        <handleSubcomponents>true</handleSubcomponents>
        <useStrictParser>false</useStrictParser>
        <useStrictValidation>false</useStrictValidation>
        <stripNamespaces>false</stripNamespaces>
        <segmentDelimiter>\r</segmentDelimiter>
        <convertLineBreaks>true</convertLineBreaks>
      </serializationProperties>
      <deserializationProperties class="com.mirth.connect.plugins.datatypes.hl7v2.HL7v2DeserializationProperties
        <useStrictParser>false</useStrictParser>
        <useStrictValidation>false</useStrictValidation>
        <segmentDelimiter>\r</segmentDelimiter>
      </deserializationProperties>
      <batchProperties class="com.mirth.connect.plugins.datatypes.hl7v2.HL7v2BatchProperties" version="4.5.0">
        <splitType>MSH_Segment</splitType>
        <batchScript></batchScript>
      </batchProperties>
      <responseGenerationProperties class="com.mirth.connect.plugins.datatypes.hl7v2.HL7v2ResponseGeneratio
        <segmentDelimiter>\r</segmentDelimiter>
        <successfulACKCode>AA</successfulACKCode>
        <successfulACKMessage></successfulACKMessage>
        <errorACKCode>AE</errorACKCode>
        <errorACKMessage>An Error Occurred Processing Message.</errorACKMessage>
        <rejectedACKCode>AR</rejectedACKCode>
        <rejectedACKMessage>Message Rejected.</rejectedACKMessage>
        <msh15ACKAccept>false</msh15ACKAccept>
        <dateFormat>yyyyMMddHHmmss.SSS</dateFormat>
      </responseGenerationProperties>
      <responseValidationProperties class="com.mirth.connect.plugins.datatypes.hl7v2.HL7v2ResponseValidationP
        <successfulACKCode>AA,CA</successfulACKCode>
        <errorACKCode>AE,CE</errorACKCode>
        <rejectedACKCode>AR,CR</rejectedACKCode>
        <validateMessageControlId>true</validateMessageControlId>
        <originalMessageControlId>Destination_Encoded</originalMessageControlId>
        <originalIdMapVariable></originalIdMapVariable>
      </responseValidationProperties>
    </inboundProperties>
    <outboundProperties class="com.mirth.connect.plugins.datatypes.hl7v2.HL7v2DataTypeProperties" version="4
      <serializationProperties class="com.mirth.connect.plugins.datatypes.hl7v2.HL7v2SerializationProperties" vers
        <handleRepetitions>true</handleRepetitions>
        <handleSubcomponents>true</handleSubcomponents>
        <useStrictParser>false</useStrictParser>
        <useStrictValidation>false</useStrictValidation>
        <stripNamespaces>false</stripNamespaces>
        <segmentDelimiter>\r</segmentDelimiter>
        <convertLineBreaks>true</convertLineBreaks>
```

3. If you select 1 of the versions and click the **Diff** button, it will open the Channel Diff window where you can see the Current version of the Channel on the left and the version you are comparing it with on the right.



Commit Id	Message
0137214f1a67ab751bcc34b5f04a431ef072c398	09/16/2025 AC - updated code..
c5670d8aa4976b73d09cc50168cfd034055df0b1	.



QA channel - Current - Editing by acervera

```
<channel version="4.5.0">
  <id>80d1004c-b1b3-463f-b589-69a120724ca3</id>
  <nextMetaDataId>2</nextMetaDataId>
  <name>QA channel</name>
  <description>this is a test 5/22</description>
  <revision>25</revision>
  <sourceConnector version="4.5.0">
    <metaDataId>0</metaDataId>
    <name>sourceConnector</name>
    <properties class="com.mirth.connect.connectors.vm.VmReceiverProperties" version="4.5.0">
      <pluginProperties/>
      <sourceConnectorProperties version="4.5.0">
        <responseVariable>None</responseVariable>
        <respondAfterProcessing>true</respondAfterProcessing>
        <processBatch>false</processBatch>
        <firstResponse>false</firstResponse>
        <processingThreads>1</processingThreads>
        <resourceIds class="linked-hash-map">
          <entry>
            <string>Default Resource</string>
            <string>[Default Resource]</string>
          </entry>
        </resourceIds>
        <queueBufferSize>1000</queueBufferSize>
      </sourceConnectorProperties>
    </properties>
    <transformer version="4.5.0">
      <elements>
        <com.mirth.connect.plugins.javascriptstep.JavaScriptStep version="4.5.0">
          <sequenceNumber>0</sequenceNumber>
          <enabled>true</enabled>
          <script>var headers = new java.util.HashMap();
// Example header (optional)
headers.put(&quot;Accept&quot;, &quot;application/json&quot;);

logger.info(&apos;test&apos;);

// SSL parameters
var sslParams = new java.util.HashMap();
sslParams.put(&quot;trustStoreName&quot;, &quot;alijks.JKS&quot;); // required
sslParams.put(&quot;keyStoreName&quot;, &quot;Ali.JKS&quot;); // required if mTLS
sslParams.put(&quot;certAlias&quot;, &quot;self signed&quot;); // required if mTLS
sslParams.put(&quot;mtlsRequired&quot;, true);
sslParams.put(&quot;hostnamePolicy&quot;, &quot;NOOP&quot;); // or &quot;STRICT&quot; for production

// HTTP parameters
var httpParams = new java.util.HashMap();
httpParams.put(&quot;connectTimeoutMs&quot;, 15000);
httpParams.put(&quot;readTimeoutMs&quot;, 30000);
httpParams.put(&quot;throwOnHttpError&quot;, false);
httpParams.put(&quot;captureHeaders&quot;, false);
```

QA channel - Time: 09-16-2025 11:16:33 - Committed by acervera

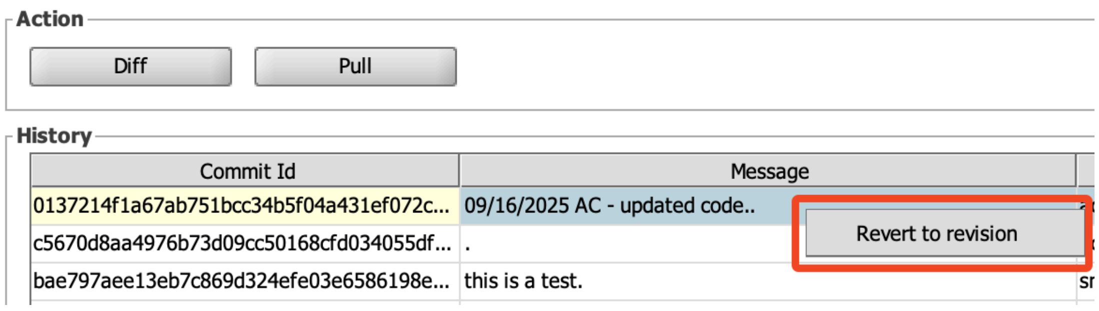
```
<channel version="4.5.0">
  <id>80d1004c-b1b3-463f-b589-69a120724ca3</id>
  <nextMetaDataId>2</nextMetaDataId>
  <name>QA channel</name>
  <description>this is a test 5/22</description>
  <revision>24</revision>
  <sourceConnector version="4.5.0">
    <metaDataId>0</metaDataId>
    <name>sourceConnector</name>
    <properties class="com.mirth.connect.connectors.vm.VmReceiverProperties" version="4.5.0">
      <pluginProperties/>
      <sourceConnectorProperties version="4.5.0">
        <responseVariable>None</responseVariable>
        <respondAfterProcessing>true</respondAfterProcessing>
        <processBatch>false</processBatch>
        <firstResponse>false</firstResponse>
        <processingThreads>1</processingThreads>
        <resourceIds class="linked-hash-map">
          <entry>
            <string>Default Resource</string>
            <string>[Default Resource]</string>
          </entry>
        </resourceIds>
        <queueBufferSize>1000</queueBufferSize>
      </sourceConnectorProperties>
    </properties>
    <transformer version="4.5.0">
      <elements>
        <com.mirth.connect.plugins.javascriptstep.JavaScriptStep version="4.5.0">
          <sequenceNumber>0</sequenceNumber>
          <enabled>true</enabled>
          <script>var headers = new java.util.HashMap();
// Example header (optional)
headers.put(&quot;Accept&quot;, &quot;application/json&quot;);

// SSL parameters
var sslParams = new java.util.HashMap();
sslParams.put(&quot;trustStoreName&quot;, &quot;alijks.JKS&quot;); // required
sslParams.put(&quot;keyStoreName&quot;, &quot;Ali.JKS&quot;); // required if mTLS
sslParams.put(&quot;certAlias&quot;, &quot;self signed&quot;); // required if mTLS
sslParams.put(&quot;mtlsRequired&quot;, true);
sslParams.put(&quot;hostnamePolicy&quot;, &quot;NOOP&quot;); // or &quot;STRICT&quot; for production

// HTTP parameters
var httpParams = new java.util.HashMap();
httpParams.put(&quot;connectTimeoutMs&quot;, 15000);
httpParams.put(&quot;readTimeoutMs&quot;, 30000);
httpParams.put(&quot;throwOnHttpError&quot;, false);
httpParams.put(&quot;captureHeaders&quot;, false);
httpParams.put(&quot;followRedirects&quot;, true);

```

Remote Repository Features

Retrieving channel revision

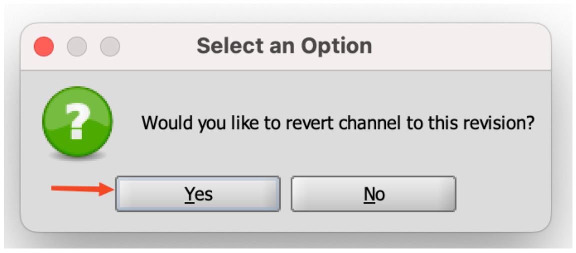
Retrieving channel revision from repo refers to the process of fetching the latest version or specific revision of a channel stored in the remote repository.

1. Within your channel, right-click on the commit you wish to revert to and select **Revert Revision.**

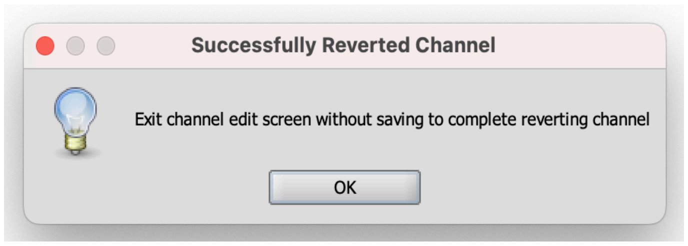


Commit Id	Message
0137214f1a67ab751bcc34b5f04a431ef072c...	09/16/2025 AC - updated code..
c5670d8aa4976b73d09cc50168cfd034055df...	.
bae797aee13eb7c869d324efe03e6586198e...	this is a test.

2. Select **Yes** to revert the channel to the selected revision. Then press **OK** to exit the channel edit screen.



Would you like to revert channel to this revision?

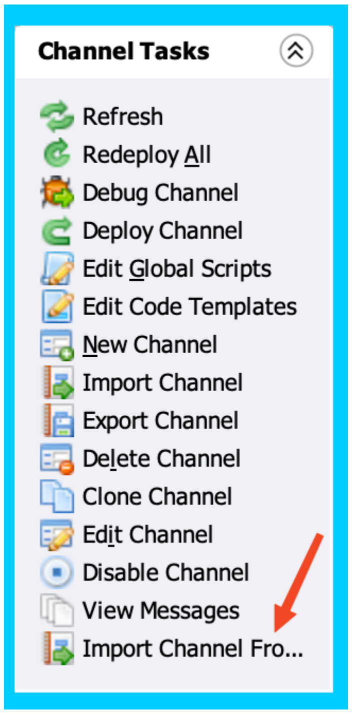


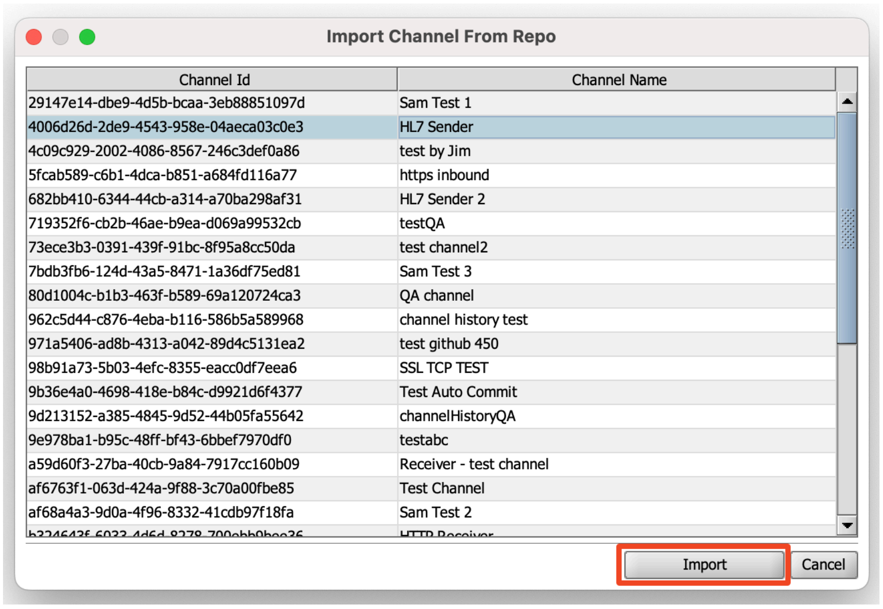
Exit channel edit screen without saving to complete reverting channel

Import a Channel

You can import channels into BridgeLink directly from the repository.

1. On the Channels Tab and under the Channel Tasks panel, select **Import Channel from Repo**. A list with the available channels to import from your repository will appear and you can select which one you want to import.



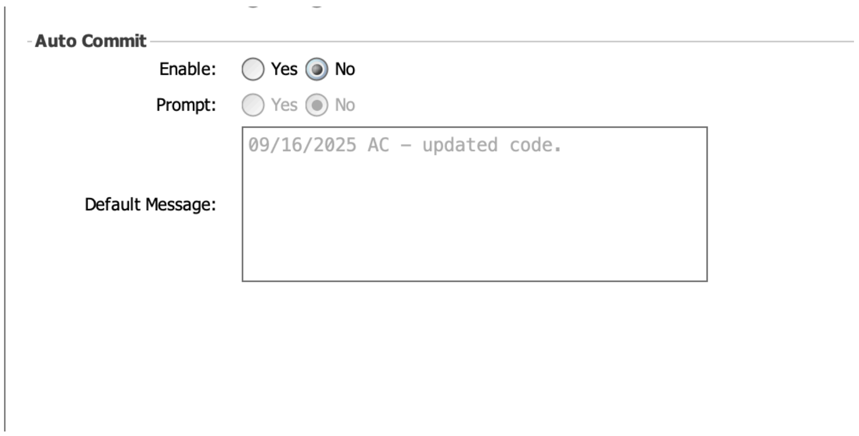


Channel Id	Channel Name
29147e14-dbe9-4d5b-bcaa-3eb88851097d	Sam Test 1
4006d26d-2de9-4543-958e-04aeca03c0e3	HL7 Sender
4c09c929-2002-4086-8567-246c3def0a86	test by Jim
5fcab589-c6b1-4dca-b851-a684fd116a77	https inbound
682bb410-6344-44cb-a314-a70ba298af31	HL7 Sender 2
719352f6-cb2b-46ae-b9ea-d069a99532cb	testQA
73ece3b3-0391-439f-91bc-8f95a8cc50da	test channel2
7bdb3fb6-124d-43a5-8471-1a36df75ed81	Sam Test 3
80d1004c-b1b3-463f-b589-69a120724ca3	QA channel
962c5d44-c876-4eba-b116-586b5a589968	channel history test
971a5406-ad8b-4313-a042-89d4c5131ea2	test github 450
98b91a73-5b03-4efc-8355-eacc0df7eea6	SSL TCP TEST
9b36e4a0-4698-418e-b84c-d9921d6f4377	Test Auto Commit
9d213152-a385-4845-9d52-44b05fa55642	channelHistoryQA
9e978ba1-b95c-48ff-bf43-6bbef7970df0	testabc
a59d60f3-27ba-40cb-9a84-7917cc160b09	Receiver - test channel
af6763f1-063d-424a-9f88-3c70a00fbe85	Test Channel
af68a4a3-9d0a-4f96-8332-41cdb97f18fa	Sam Test 2

2. Click **Import** and save the Channel Settings.

Manual/Auto Commit

In Channel History Settings, there is an **Auto Commit** option.



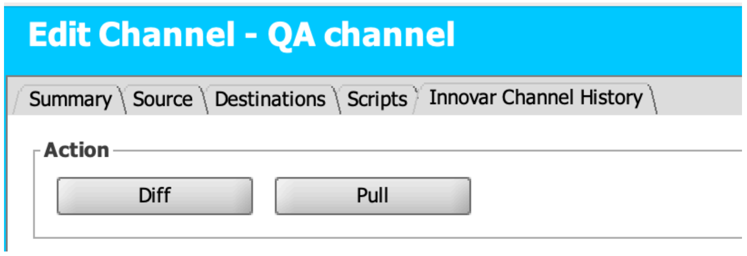
Auto Commit

Enable: Yes [✓ No]

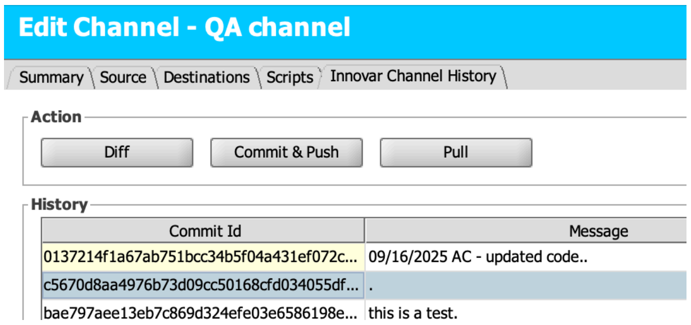
Prompt: Yes [✓ No]

Default Message: 09/16/2025 AC – updated code.

- When Auto Commit is enabled, the Commit & Push action will not be visible and you only will be able to see Diff and Pull.



- When Auto Commit is disabled, there will be three action options **Diff**, **Commit & Push**, and **Pull** under the **Channel History** tab on each of your channels.



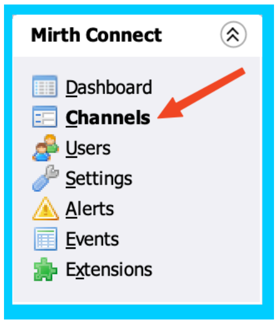
Commit Id	Message
0137214f1a67ab751bcc34b5f04a431ef072c...	09/16/2025 AC - updated code..
c5670d8aa4976b73d09cc50168cfd034055df...	.
bae797aee13eb7c869d324efe03e6586198e...	this is a test.

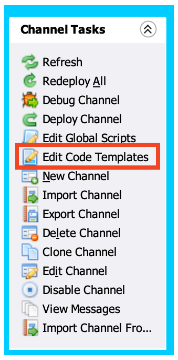
Code Templates

View history on Code templates

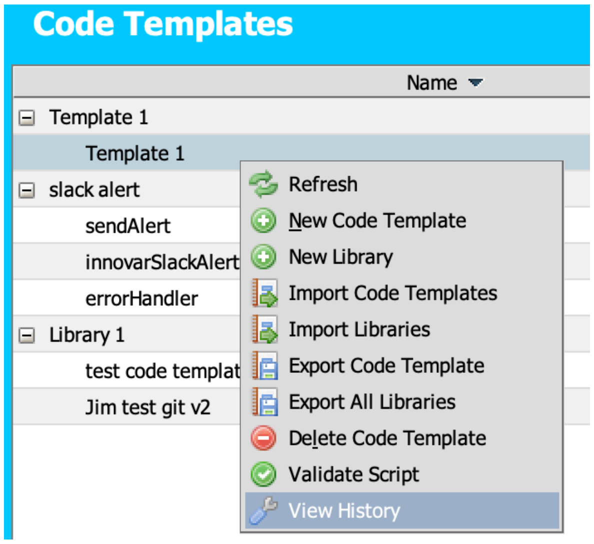
We can also review the code template history. Start by modifying the target code template and saving it.

1. Click on **Channels** and select **Edit Code Templates** in the Channel Tasks.

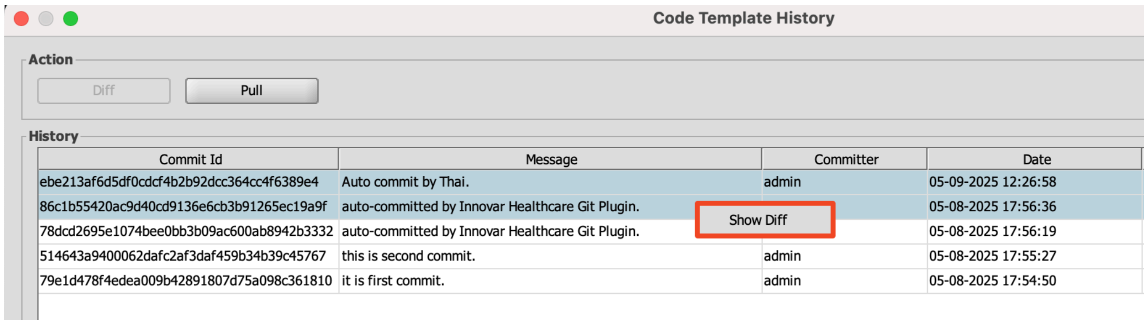




2. Highlight the code template that you want to check the History for, right-click, and select **View History**.

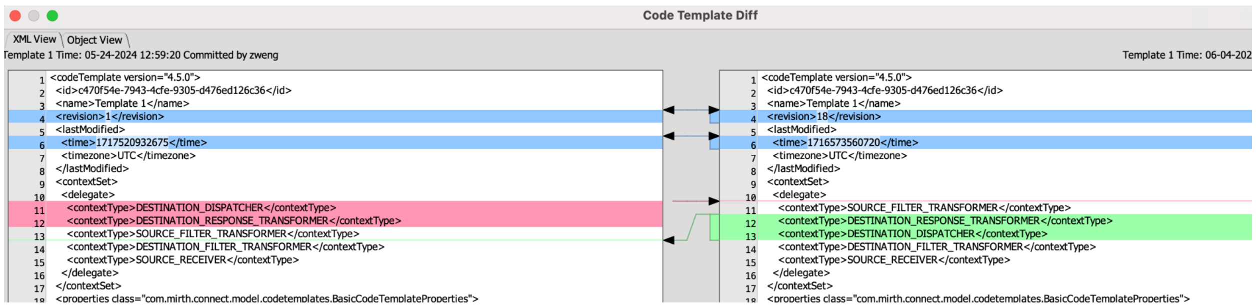


3. Select two commits in the Code Template History window, right-click, and choose Show Diff.



Commit Id	Message	Committer	Date
ebe213af6d5df0cdcf4b2b92dcc364cc4f6389e4	Auto commit by Thai.	admin	05-09-2025 12:26:58
86c1b55420ac9d40cd9136e6cb3b91265ec19a9f	auto-committed by Innovar Healthcare Git Plugin.		05-08-2025 17:56:36
78dcd2695e1074bee0bb3b09ac600ab8942b3332	auto-committed by Innovar Healthcare Git Plugin.		05-08-2025 17:56:19
514643a9400062dafc2af3daf459b34b39c45767	this is second commit.	admin	05-08-2025 17:55:27
79e1d478f4edea009b42891807d75a098c361810	it is first commit.	admin	05-08-2025 17:54:50

4. The Diff window displays the differences between the two selected commits.



Template 1 Time: 05-24-2024 12:59:20 Committed by zweng

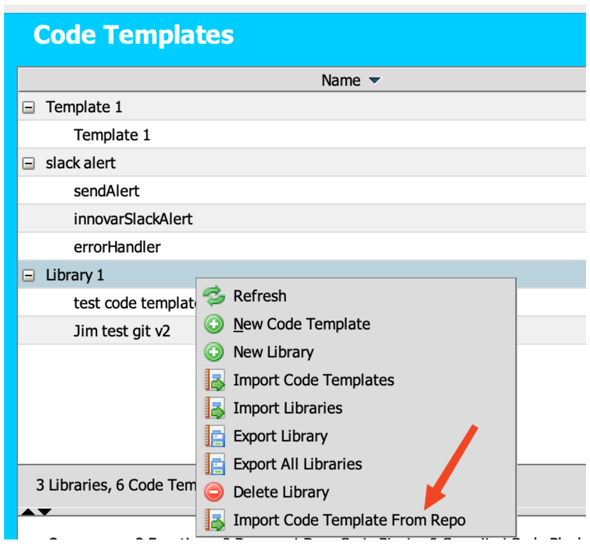
```
<codeTemplate version="4.5.0">
  <id>c470f54e-7943-4cfe-9305-d476ed126c36</id>
  <name>Template 1</name>
  <revision>1</revision>
  <lastModified>
    <time>1717520932675</time>
    <timezone>UTC</timezone>
  </lastModified>
  <contextSet>
    <delegate>
      <contextType>DESTINATION_DISPATCHER</contextType>
      <contextType>DESTINATION_RESPONSE_TRANSFORMER</contextType>
      <contextType>SOURCE_FILTER_TRANSFORMER</contextType>
      <contextType>DESTINATION_FILTER_TRANSFORMER</contextType>
      <contextType>SOURCE_RECEIVER</contextType>
    </delegate>
  </contextSet>
  <properties class="com.mirth.connect.model.codetemplates.BasicCodeTemplateProperties">
```

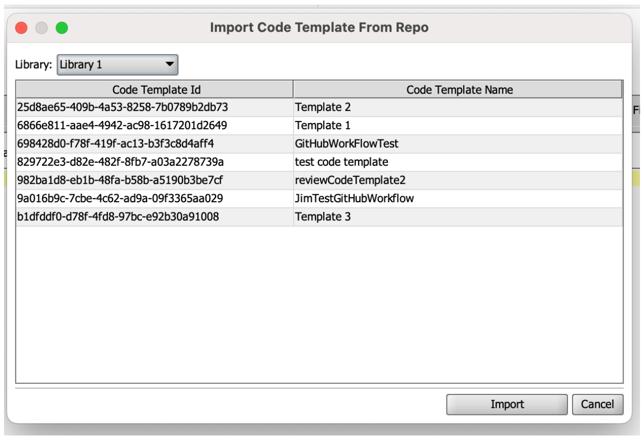
Template 1 Time: 06-04-202

```
<codeTemplate version="4.5.0">
  <id>c470f54e-7943-4cfe-9305-d476ed126c36</id>
  <name>Template 1</name>
  <revision>18</revision>
  <lastModified>
    <time>1716573560720</time>
    <timezone>UTC</timezone>
  </lastModified>
  <contextSet>
    <delegate>
      <contextType>SOURCE_FILTER_TRANSFORMER</contextType>
      <contextType>DESTINATION_RESPONSE_TRANSFORMER</contextType>
      <contextType>DESTINATION_DISPATCHER</contextType>
      <contextType>DESTINATION_FILTER_TRANSFORMER</contextType>
      <contextType>SOURCE_RECEIVER</contextType>
    </delegate>
  </contextSet>
  <properties class="com.mirth.connect.model.codetemplates.BasicCodeTemplateProperties">
```

Import code templates

1. Right click on a library and select **Import Code Template From Repo.**

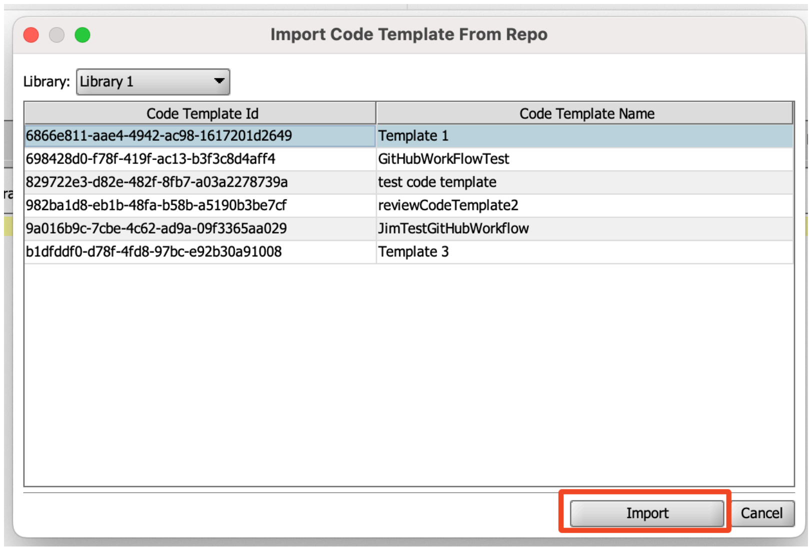




Library: Library 1

Code Template Id	Code Template Name
25d8ae65-409b-4a53-8258-7b0789b2db73	Template 2
6866e811-aae4-4942-ac98-1617201d2649	Template 1
698428d0-f78f-419f-ac13-b3f3c8d4aff4	GitHubWorkFlowTest
829722e3-d82e-482f-8fb7-a03a2278739a	test code template
982ba1d8-eb1b-48fa-b58b-a5190b3be7cf	reviewCodeTemplate2
9a016b9c-7cbe-4c62-ad9a-09f3365aa029	JimTestGitHubWorkflow
b1dfddf0-d78f-4fd8-97bc-e92b30a91008	Template 3

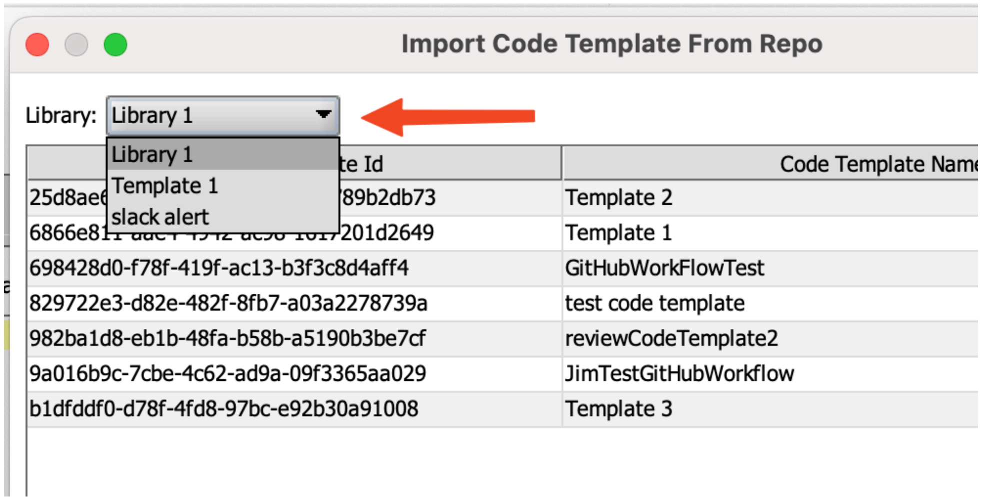
2. Choose the code template you want to import from the list and click **Import**.



Library: Library 1

Code Template Id	Code Template Name
6866e811-aae4-4942-ac98-1617201d2649	Template 1
698428d0-f78f-419f-ac13-b3f3c8d4aff4	GitHubWorkFlowTest
829722e3-d82e-482f-8fb7-a03a2278739a	test code template
982ba1d8-eb1b-48fa-b58b-a5190b3be7cf	reviewCodeTemplate2
9a016b9c-7cbe-4c62-ad9a-09f3365aa029	JimTestGitHubWorkflow
b1dfddf0-d78f-4fd8-97bc-e92b30a91008	Template 3

3. The imported Code Template will appear under the selected library. Also you can select which library you want to import it on the dropdown button.


Import Code Template From Repo

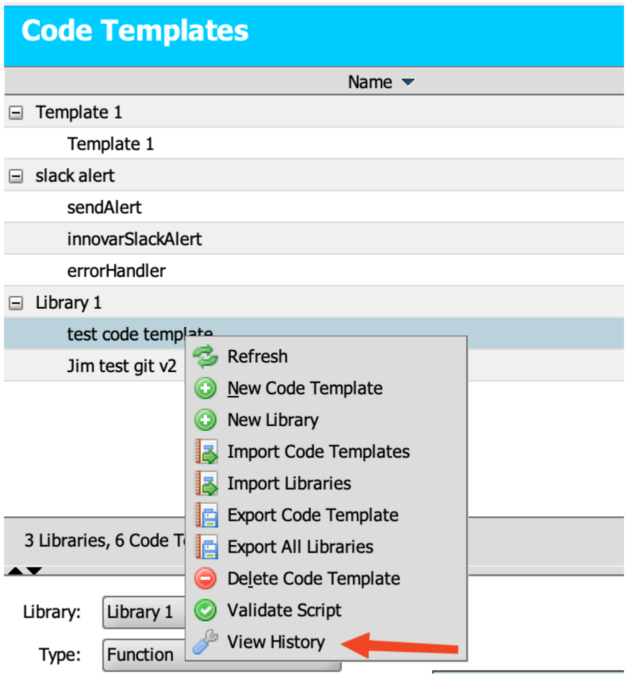
Library: Library 1

Library 1
Template 1
slack alert

te Id	Code Template Name
25d8ae6 …789b2db73	Template 2
6866e81… 201d2649	Template 1
698428d0-f78f-419f-ac13-b3f3c8d4aff4	GitHubWorkFlowTest
829722e3-d82e-482f-8fb7-a03a2278739a	test code template
982ba1d8-eb1b-48fa-b58b-a5190b3be7cf	reviewCodeTemplate2
9a016b9c-7cbe-4c62-ad9a-09f3365aa029	JimTestGitHubWorkflow
b1dfddf0-d78f-4fd8-97bc-e92b30a91008	Template 3

Revert code template changes

1. Right click on the template and click **View History.**


Code Templates

Name

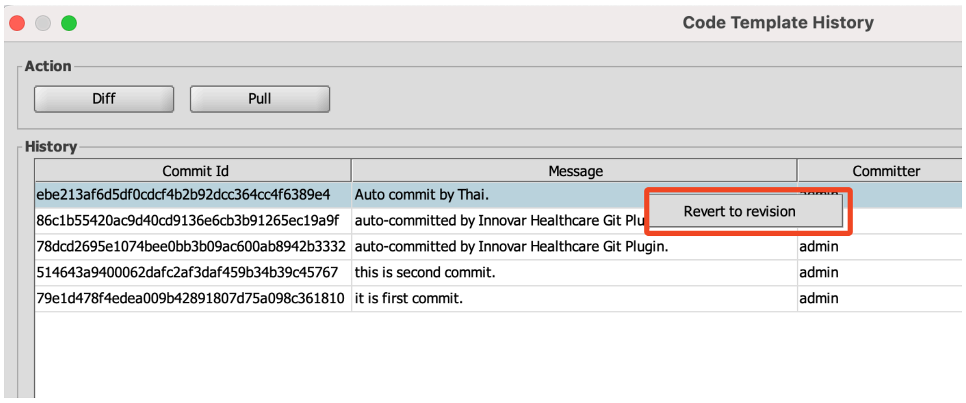
- Template 1
  - Template 1
- slack alert
  - sendAlert
  - innovarSlackAlert
  - errorHandler
- Library 1
  - test code template
  - Jim test git v2

3 Libraries, 6 Code T…

Library: Library 1

Type: Function

2. Right click on the desired commit and press **Revert to revision.**


Code Template History

Action

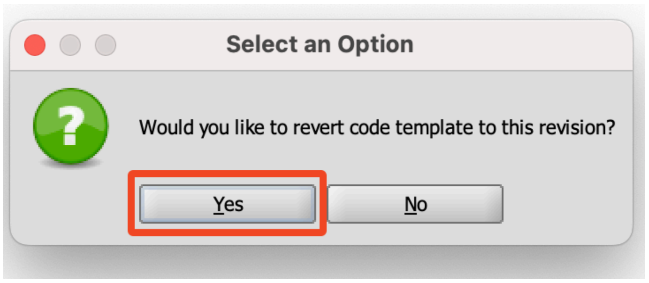
Diff Pull

History

Commit Id	Message	Committer
ebe213af6d5df0cdcf4b2b92dcc364cc4f6389e4	Auto commit by Thai.	
86c1b55420ac9d40cd9136e6cb3b91265ec19a9f	auto-committed by Innovar Healthcare Git Plu	
78dcd2695e1074bee0bb3b09ac600ab8942b3332	auto-committed by Innovar Healthcare Git Plugin.	admin
514643a9400062dafc2af3daf459b34b39c45767	this is second commit.	admin
79e1d478f4edea009b42891807d75a098c361810	it is first commit.	admin

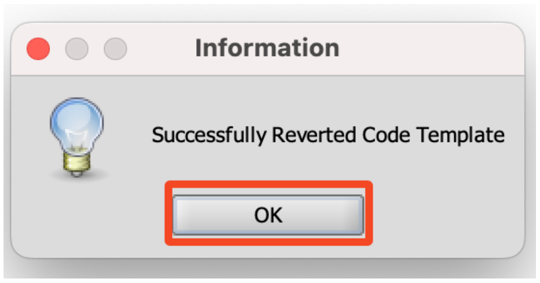
Revert to revision

3. Press **Yes** and the Code Template will be reverted to the version selected.


Select an Option

Would you like to revert code template to this revision?

Yes No


Information

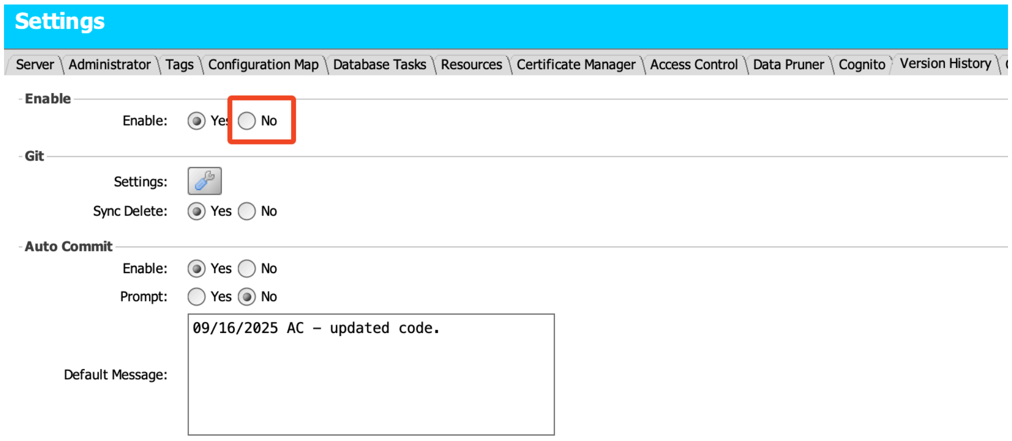
Successfully Reverted Code Template

OK

Disabling the Channel History Plugin in BridgeLink

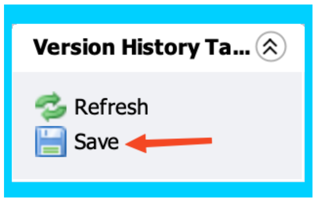
Sometimes, committing and pushing every change to the remote repository might not be necessary. This section explains how to disable the Channel History plugin.

1. Navigate to the settings and click on the Version History tab and under Enable field, select **No**.

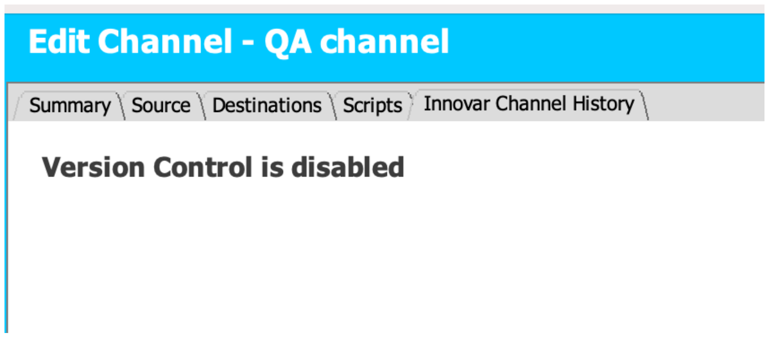


Enable
Enable: [✓ Yes] No
Git
Settings:
Sync Delete: [✓ Yes] No
Auto Commit
Enable: [✓ Yes] No
Prompt: Yes [✓ No]
Default Message: 09/16/2025 AC – updated code.

2. Click **Save** on the task panel to disable the plugin.



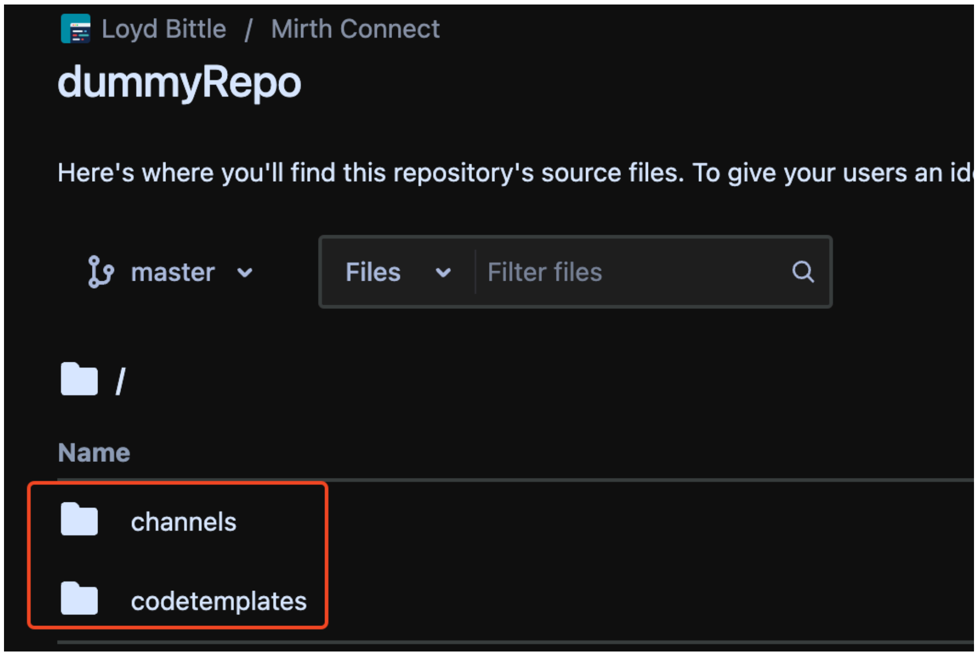
3. Now you will see this message under the Innovar Channel History tab on your channels.



Version Control is disabled

View the code history on remote repository

The Channel History plugin pushes every new commit to your remote repository. The XML files for channels and code templates are stored in their respective folders, with the XML file names corresponding to either the channel ID or the code template ID.



Loyd Bittle / Mirth Connect

dummyRepo

Here's where you'll find this repository's source files. To give your users an id