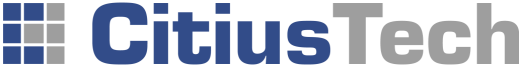


CitiusTech User Guide

NextGen Connect GIT Plugin for Version Control

Version 1.0



Copyright

This document is CitiusTech Internal and contains proprietary information, including trade secrets of CitiusTech. Neither the document nor any of the information contained in it may be reproduced or disclosed to any unauthorized person under any circumstances without the express written permission of CitiusTech.

Intended Audience

This document is intended for the users of CitiusTech’s NextGen (previously known as Mirth) Connect GIT plugins developed for version control using GIT libraries.

Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Document Version # | Revision Date | Prepared By | Approved By | Approval Date | Summary of Changes |
| 1.0 | 29-07-2020 | Pravin Gadade |  | Click here to enter a date. | First Version |
| 1.1 | Click here to enter a date. |  |  | Click here to enter a date. |  |
| 1.2 | Click here to enter a date. |  |  | Click here to enter a date. |  |

How to Use this Document (Optional)

This user guide provides a suggested workflow and step-by step walkthrough of the steps on NextGen Connect integration interface in order to maintain the version control of Mirth channels. Use this document as a guide to log in to NextGen Connect servers and use GIT plugin developed for version control.

Before you Begin

To use this guide successfully, end-users must have :

1. GIT plugin files installed on the NextGen Connect Integration engine (Please refer to GIT Plugin Installation guide for more details on this)
2. Mirth Connect application (Preferably version 3.9 and above) shall be installed on the system accessing the plugin
3. Java (Java 8 Onwards) shall be installed on system accessing the plugins
4. Valid link and Valid login credentials to access the NextGen Connect integration server instance
5. Valid New code signing certificate for signing the .jar files
6. GIT repository URL for maintaining the .xml files pertaining to each version of the Channels.
7. .exe files for custom editor / code comparing tool (Optional)

Acronyms and Abbreviations

This section defines the acronyms and abbreviations used in the document.

|  |  |
| --- | --- |
| Term | Definition |
| Instance | This term identifies the NextGen Connect server instance |
| Channel | Channel represents the integration interface configured on each NextGen Connect server instance. There can be around 150+ interfaces configured on each of the server instance. |
| CT | CT is the short form of term ‘CitiusTech’. |

Contents

[1 Step by Step User Guide 6](#_Toc46850592)

[1.1 Logging in to NextGen Connect Server Monitoring Dashboard 6](#_Toc46850593)

[1.2 Using Mirth Metrics Dashboard 7](#_Toc46850594)

[1.3 Drill down Views: 9](#_Toc46850595)

[1.3.1 Drill down from Server Monitor Section: 9](#_Toc46850596)

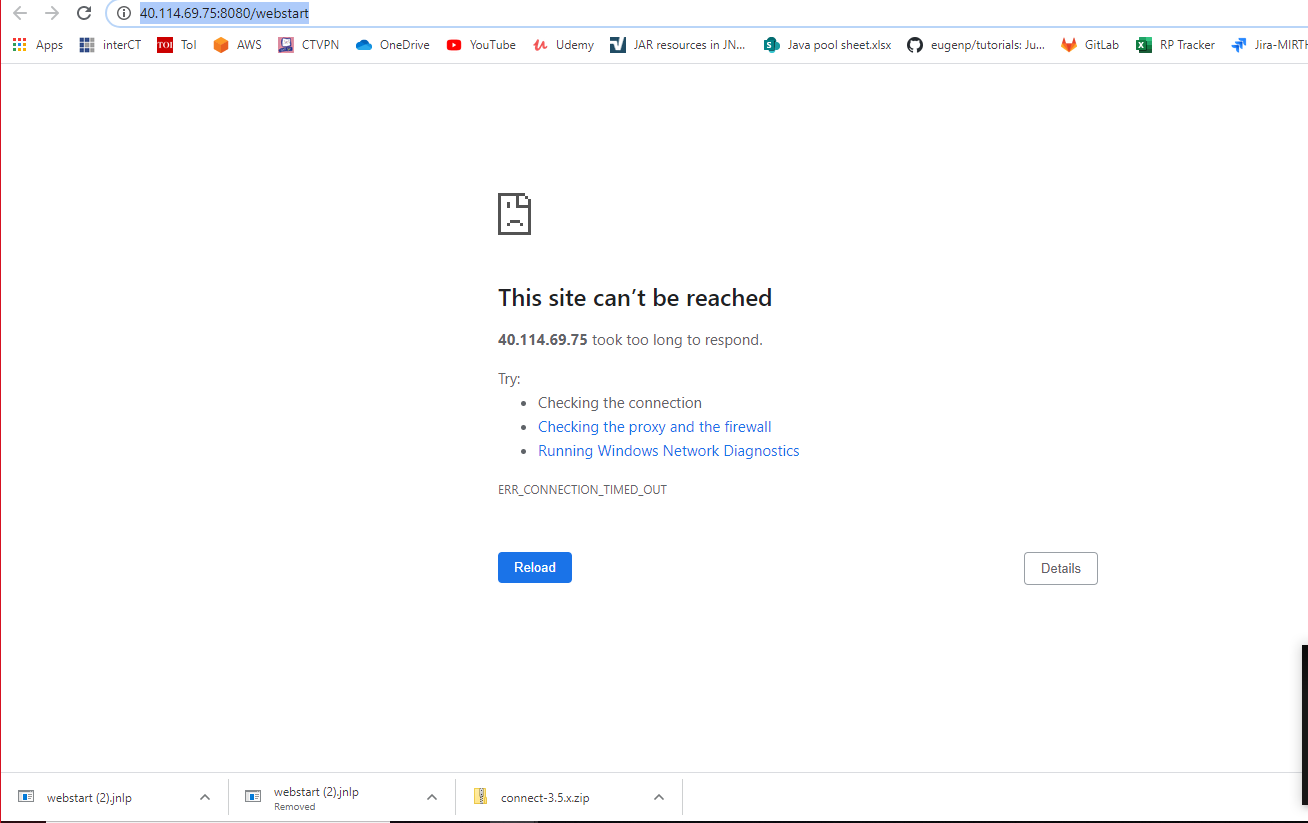
[1.3.2 Drill down from Interface/Channel Monitor Section 11](#_Toc46850597)

# Step by Step User Guide

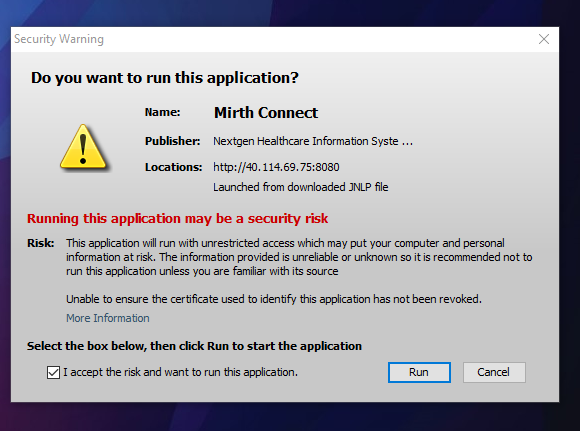
This section describes step by step guide to access the GIT Plugin and its various features on NextGen Connect Server.

## Logging in to NextGen Connect Server

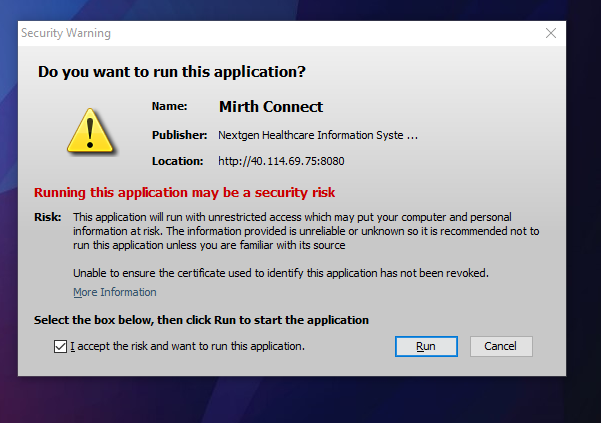
1. Click on the NextGen Connect Server link to invoke the server instance : <http://40.114.69.75:8080/webstart>
2. Opening the server instance link in browser shall download a .jnlp file in browser’s downloads section.
3. Once the .jnlp file download is completed , ‘Run’ the file.



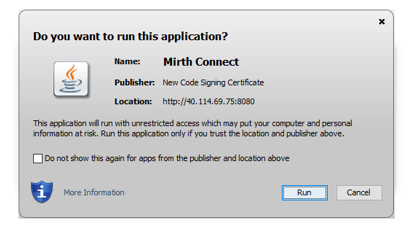
1. This shall open a pop-up as follows asking user to run the ‘Mirth Connect’ application. Click on the check box for adding the disclaimer and click on ‘Run’ button.



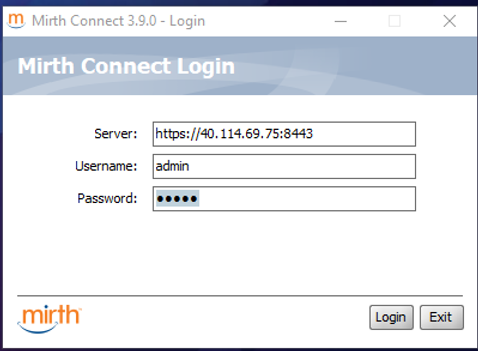
1. After selecting ‘Run’ option, one more similar pop-up box shall open asking user to run the ‘Mirth Connect’ application. Click on the check box for adding the disclaimer and click on ‘Run’ button.



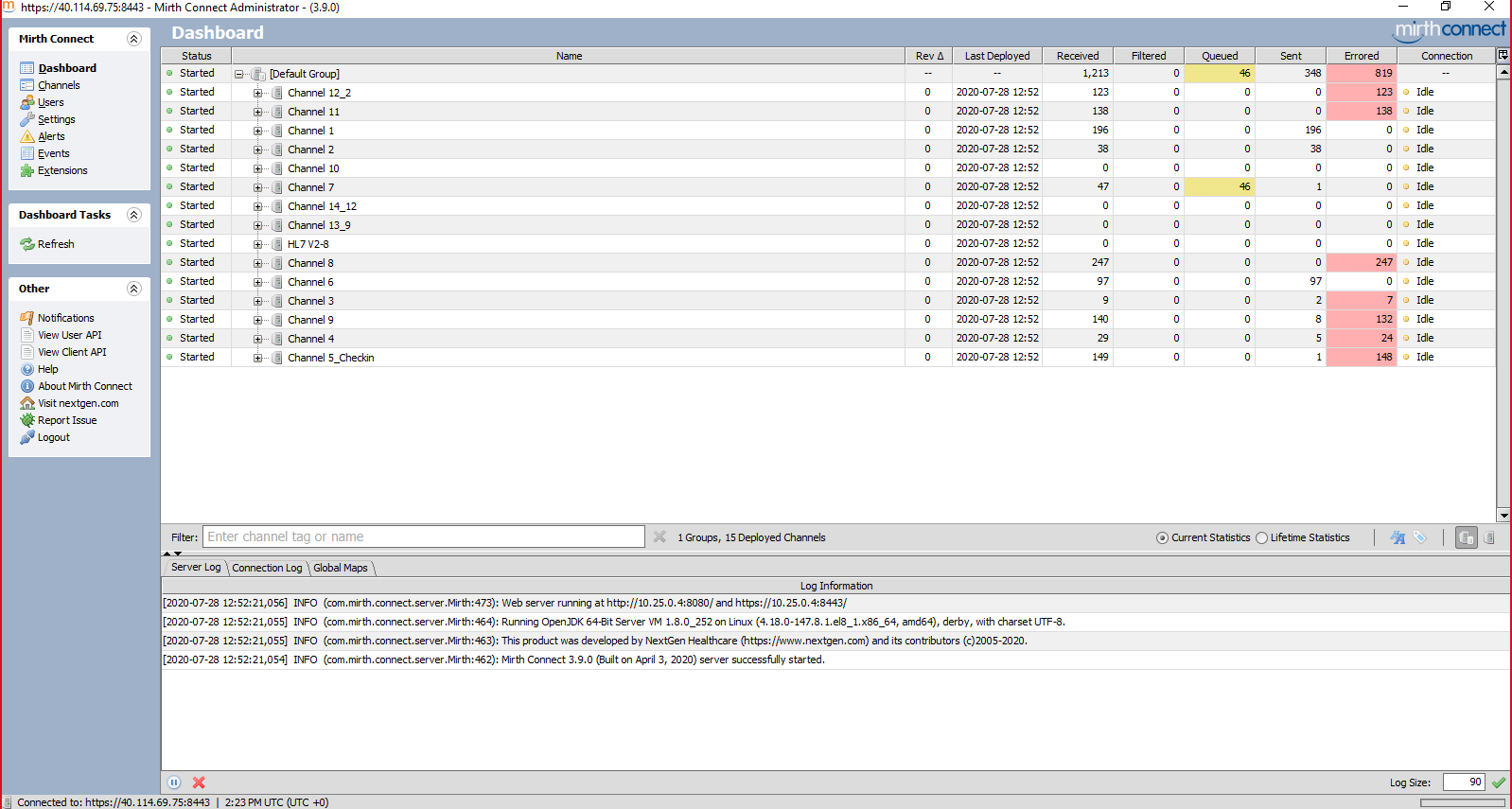
1. This shall open a pop-up message box asking to ‘Run’ the ‘New Code Signing certificate’ for Mirth Connect. Click on ‘Run’ button.



1. This shall open the login video for logging in to Mirth Connect. Enter the valid username and password to login to the Mirth Connect.



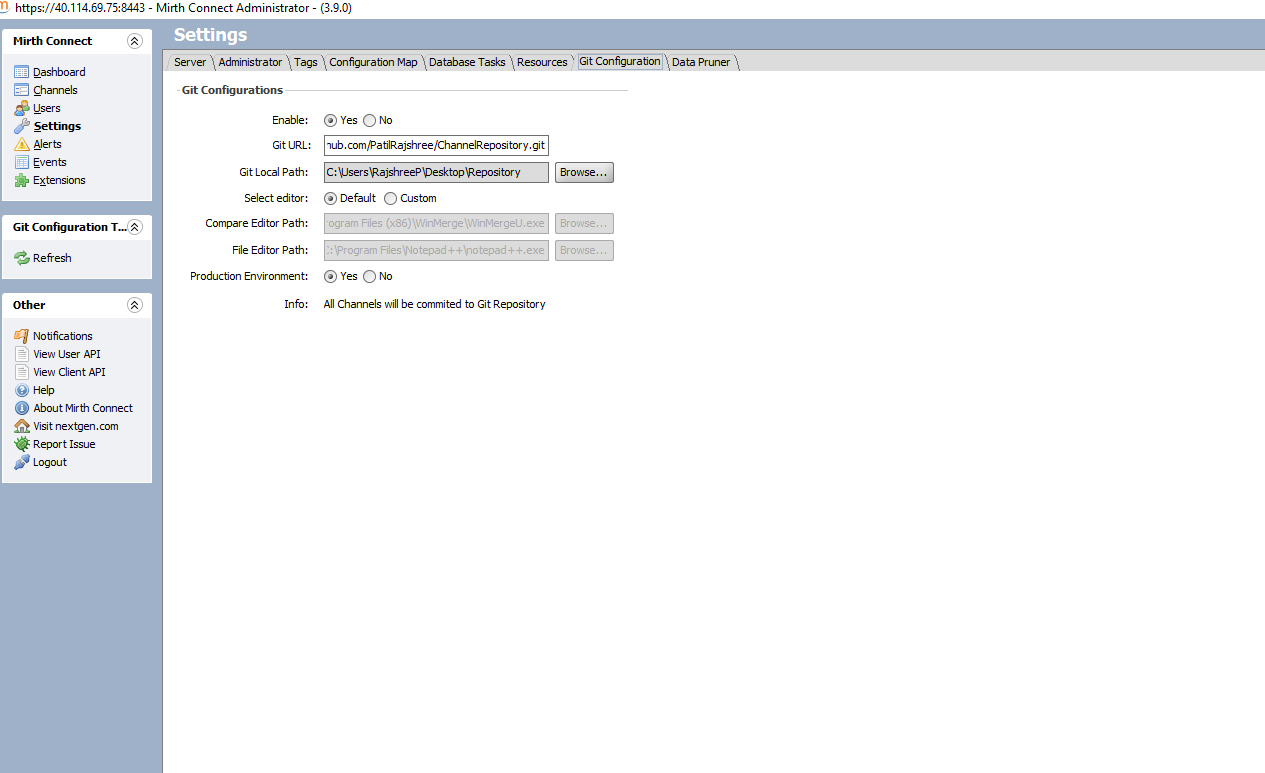
1. This shall navigate user to the Mirth Connect Dashboard screen. Click on ‘Extensions’ link on left hand menu panel and check if Git plugin is present in Installed Plugins table. If GIT Plugin is not available get GIT Plugin installed in the system and login again to Mirth Connect. (Please refer to Installation guide for more details on GIT Plugin installation).



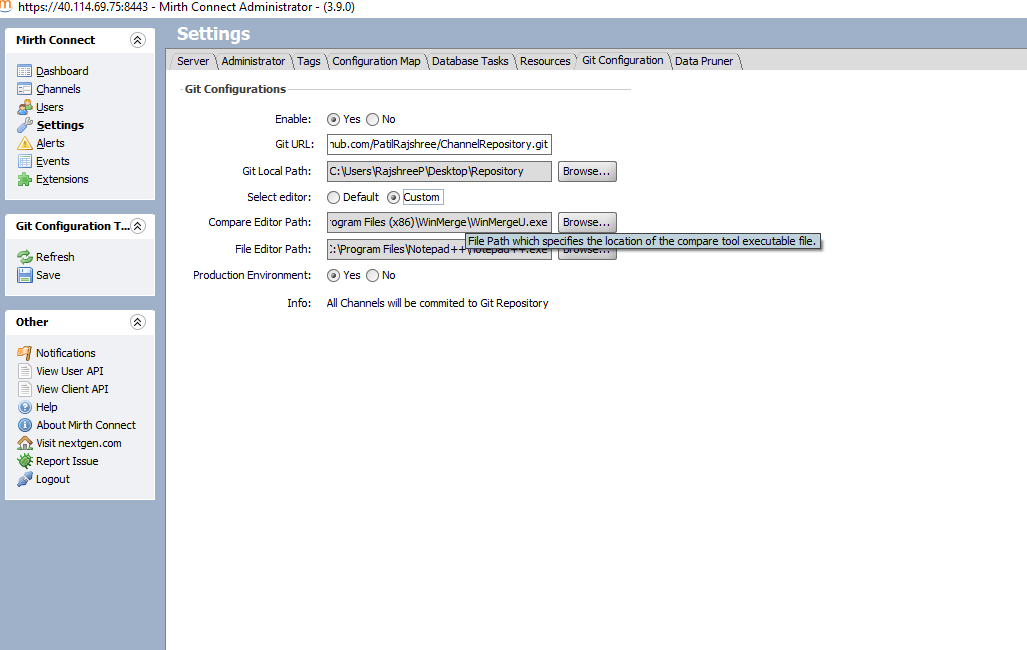
1. Once GIT Plugins are installed and can be viewed in Extensions section, click on ‘Settings’ from Left Menu Panel for configurations. At this stage ‘Git configuration’ tab should be visible in header section of the screen.

## Setting Up GIT Configuration

1. Click on the Git configuration tab where all git configurations are visible.



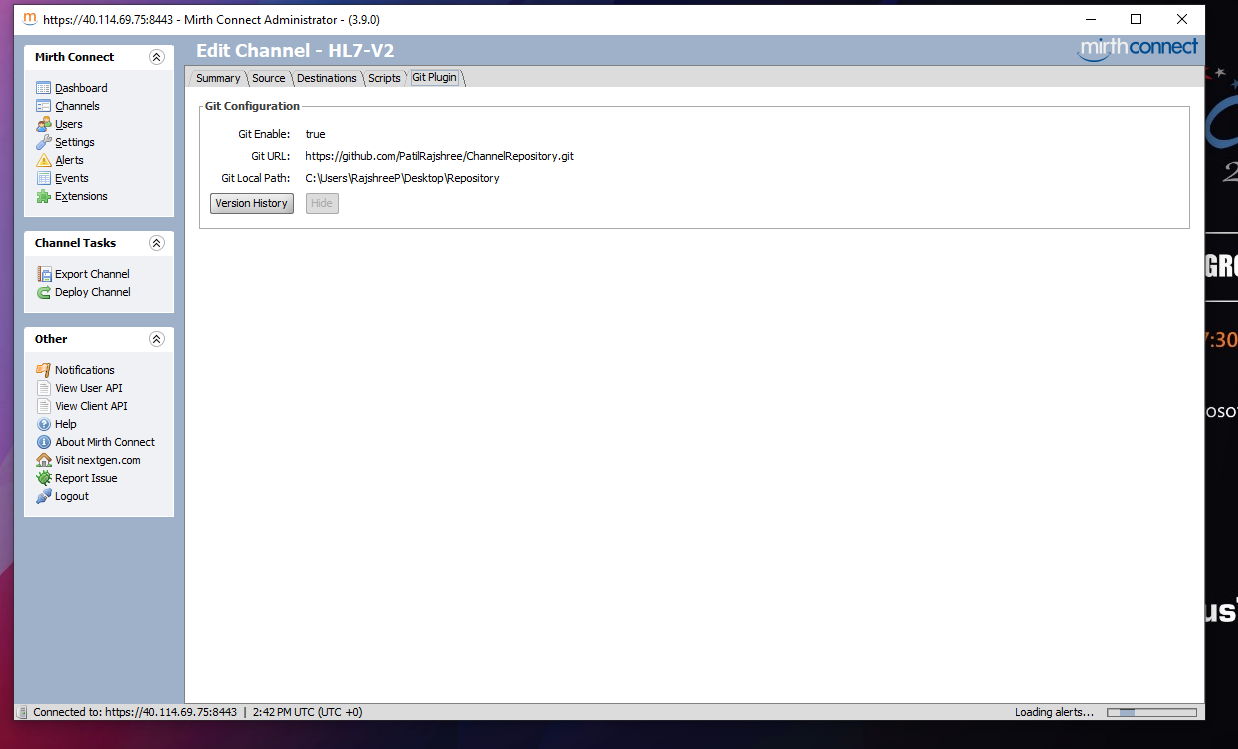
1. Select ‘Yes’ in front of ‘Enable’ attribute, to enable the GIT Plugin and its associated configurations.
2. Enter the URL of ‘GIT Repository’ where you want to store the .xml files for version control in text box in front of ‘GIT URL’ section.
3. Enter the path of the local folder on system for storing all the xml files on system by browsing your system in ‘GIT Local Path’ section.
4. Select the Editor option as per your choice. Selecting ‘Default’ option shall open , the default editor and comparator tool for viewing and comparing the xml available on Mirth Connect.
5. Selecting ‘ Custom’ option shall allow user to use Notepad++ for editing the xml files and Win-merge for comparing the xml files. If user selects the ‘Custom’ option , user need to input the file path of executable files (.exe files) in ‘Comparator Editor Path and ‘File Editor path‘ sections to use desired tools.



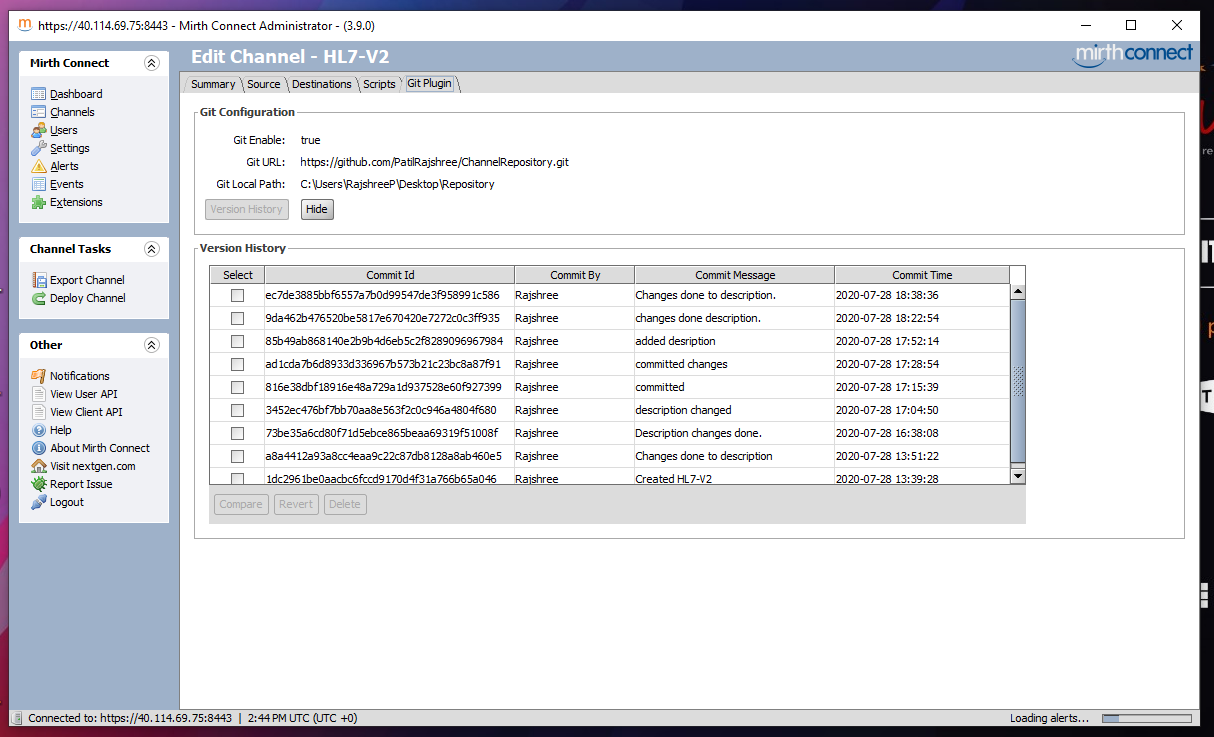
1. User need to select the ‘Yes’ or ‘No’ radio button against the ‘Production Environment’ section. If user selects the ‘No’ option , user shall see an option to save the xml files to GIT repository every time any change is made. If user selects ‘Yes’ option , all the changes made shall be by default saved to the GIT repository.
2. Once all the settings are done , click on ‘Save’ option in the ‘Git Configuration Tab’ section of the left-hand menu panel to start using the GIT plugin.

## Using GIT Plugin for Channel

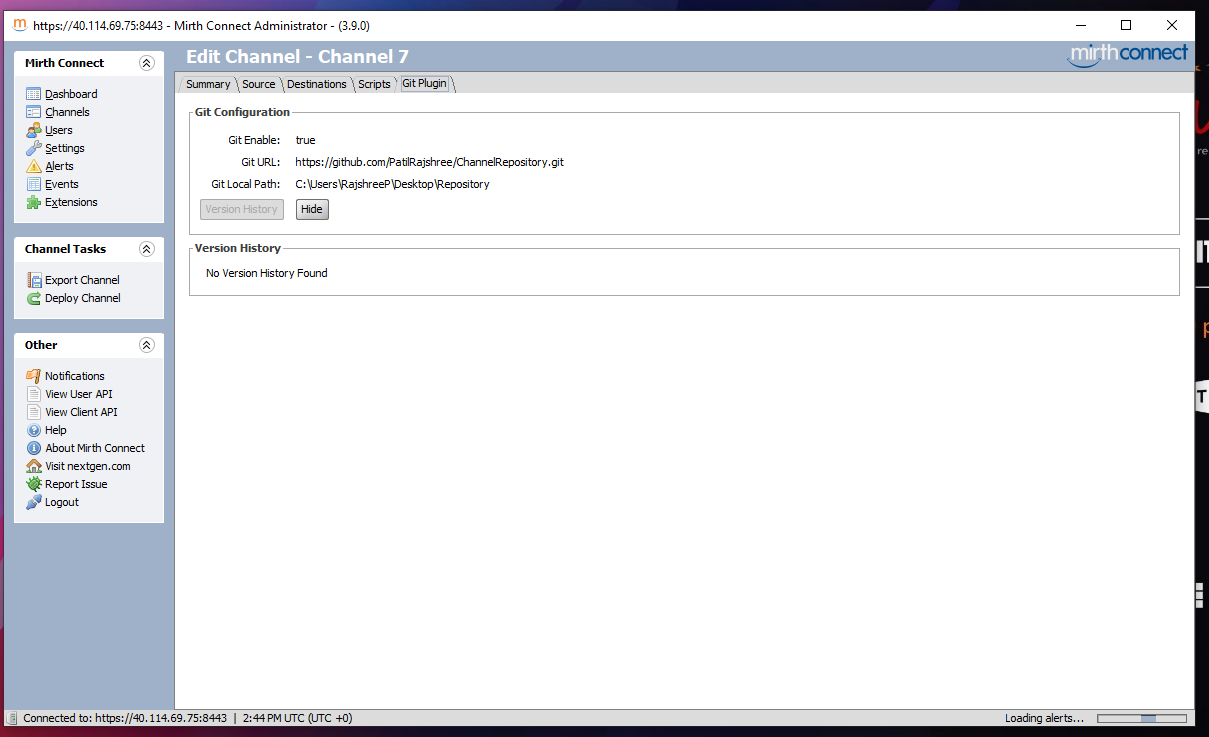
1. Once GIT Configurations are done and successfully saved, click on ‘Channels’ from ‘Mirth Connect’ section of the menu panel on left hand side.
2. On selecting a particular channel from channel list , Git Plugin tab should be visible along with the other existing tabs.
3. Click on the Git Plugin tab, where user can see all the settings configured in section 1.2 above along with the Version History and Hide button.



1. Select ‘Version History’ tab to view the list of previous versions of the channel configurations.

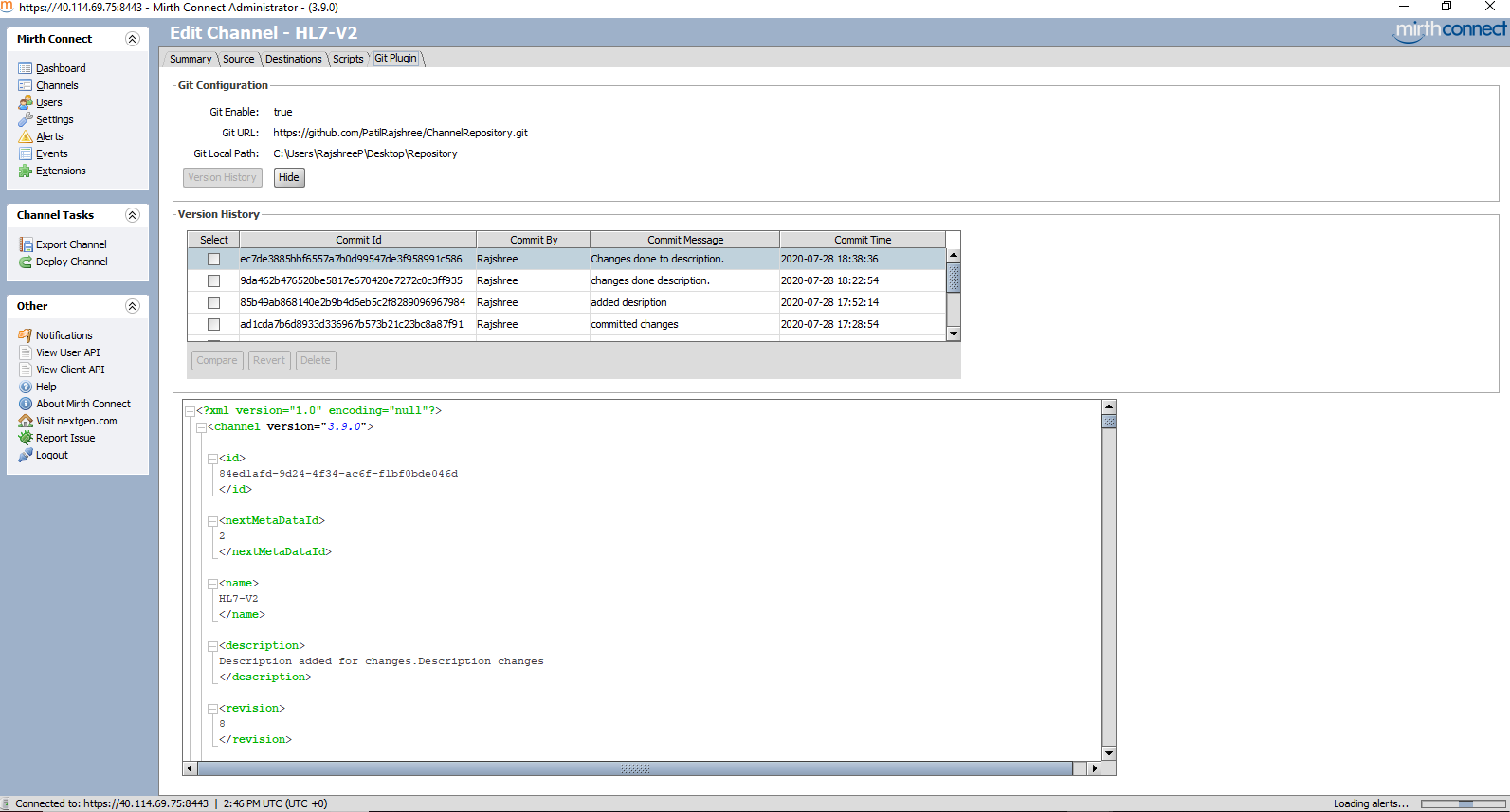


1. If there are no previous versions of configuration xml files available for the given channel, then following screen shall be seen.

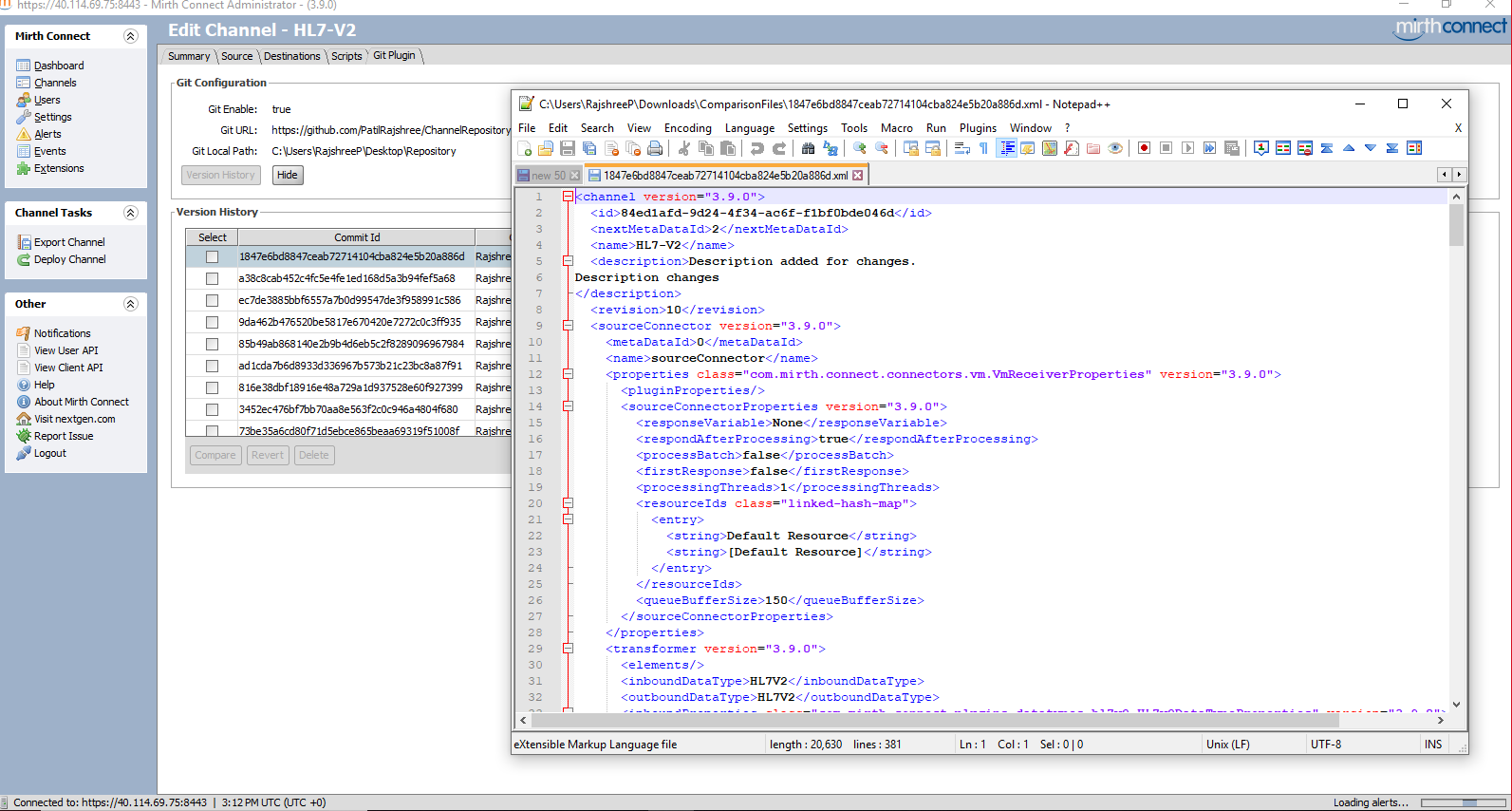


**View xml files of previous version:**

1. Double click on any of the version from the list of available versions to view the .xml file in file editor.



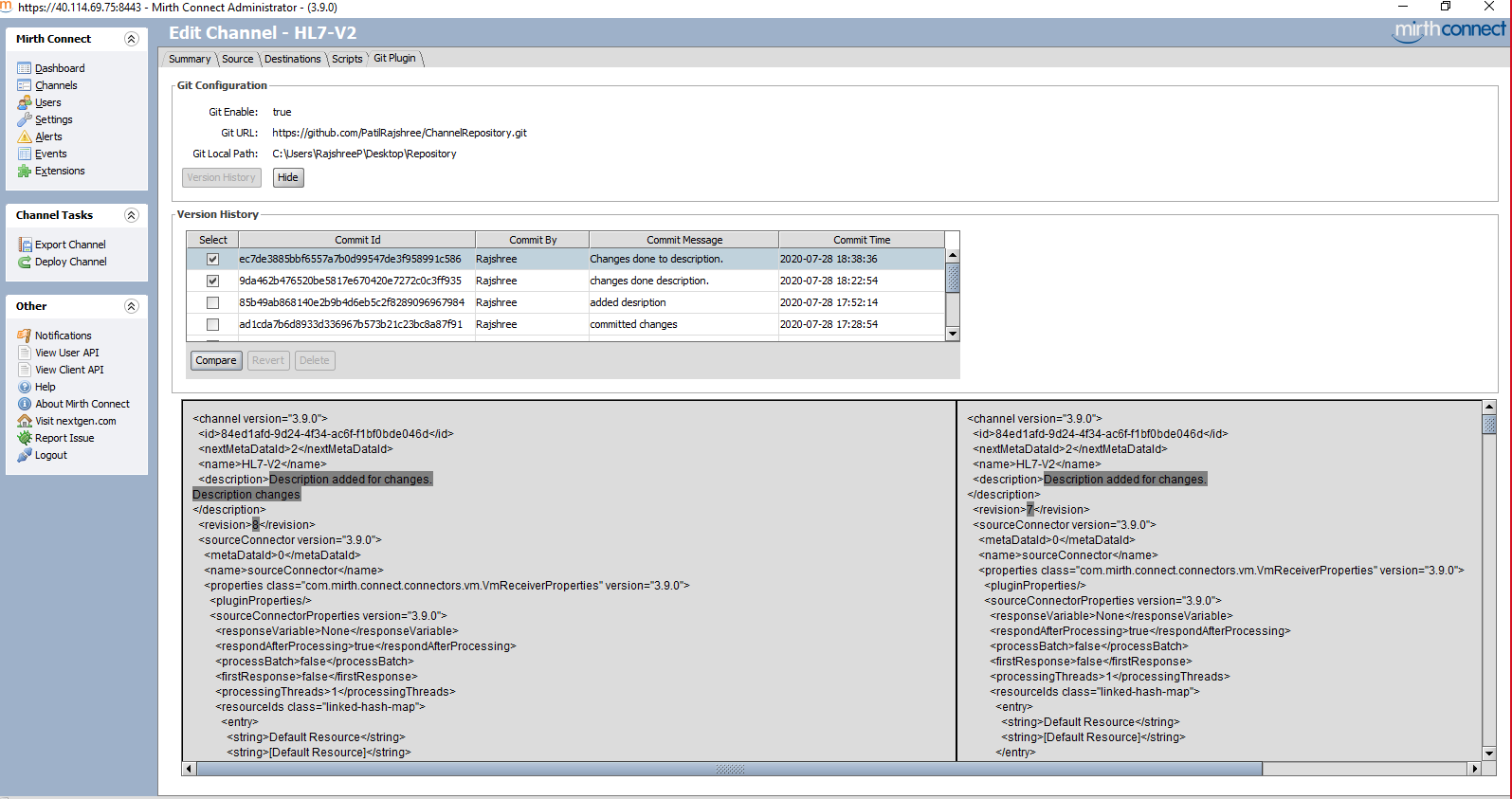
1. In case user has selected the custom file editor while making the configurations, the file shall be opened in the selected custom editor.



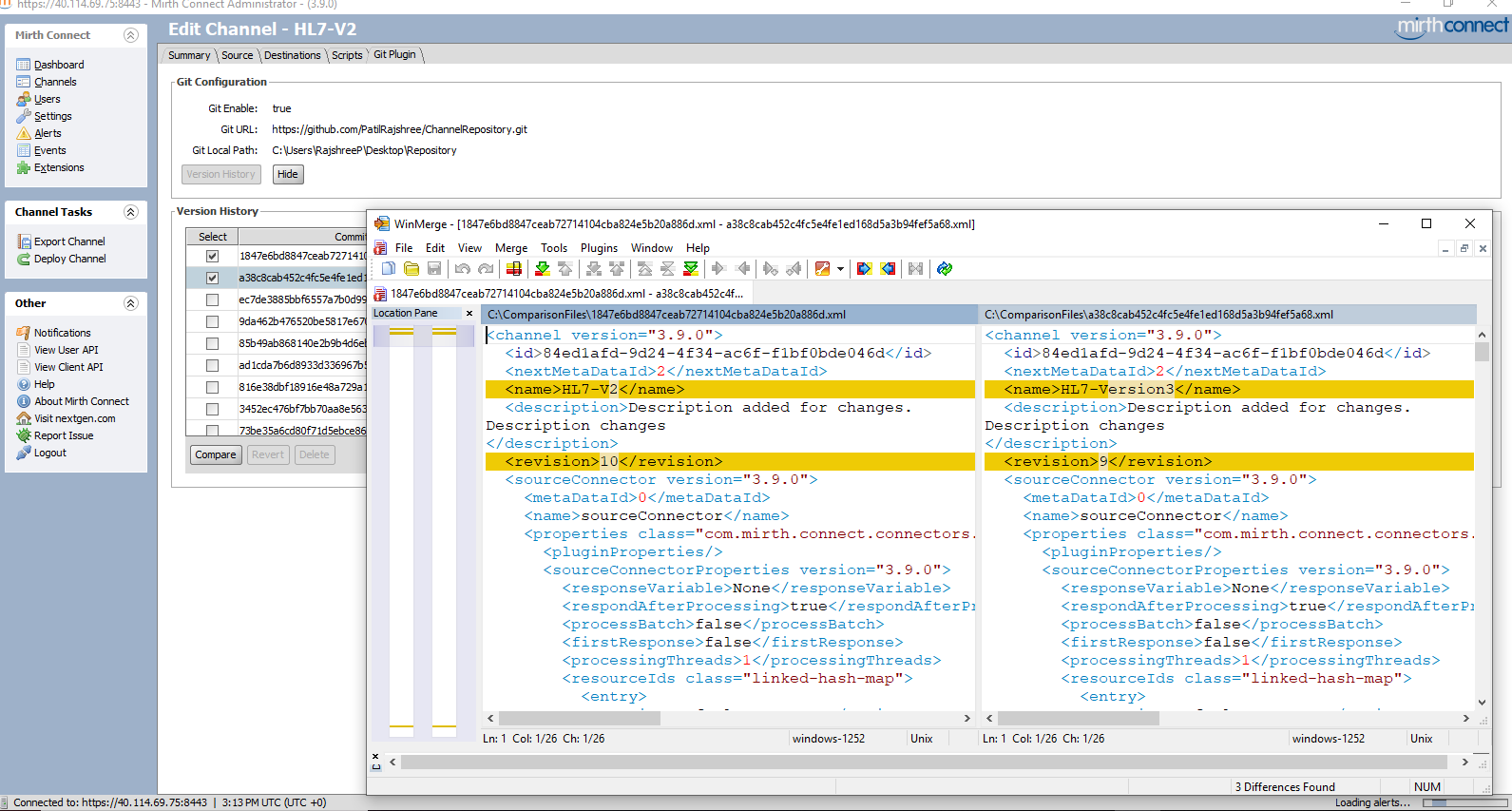
**‘Compare’ any two previous versions:**

1. To ‘Compare’ any two versions , select the checkboxes against the desired versions and select ‘Compare’ button.

**Default Mirth Comparator**:

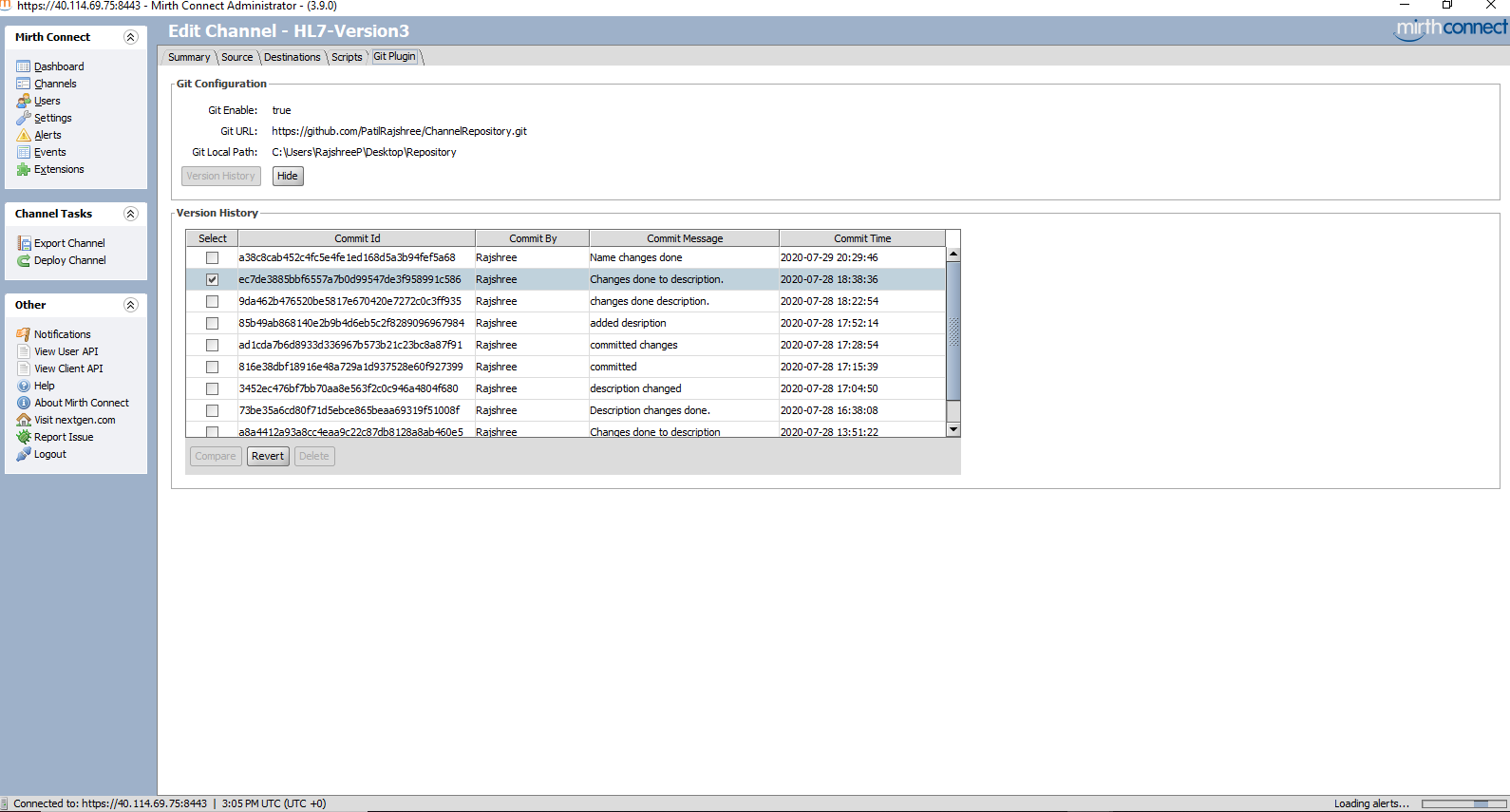


**Custom Win-Merge Comparator**:

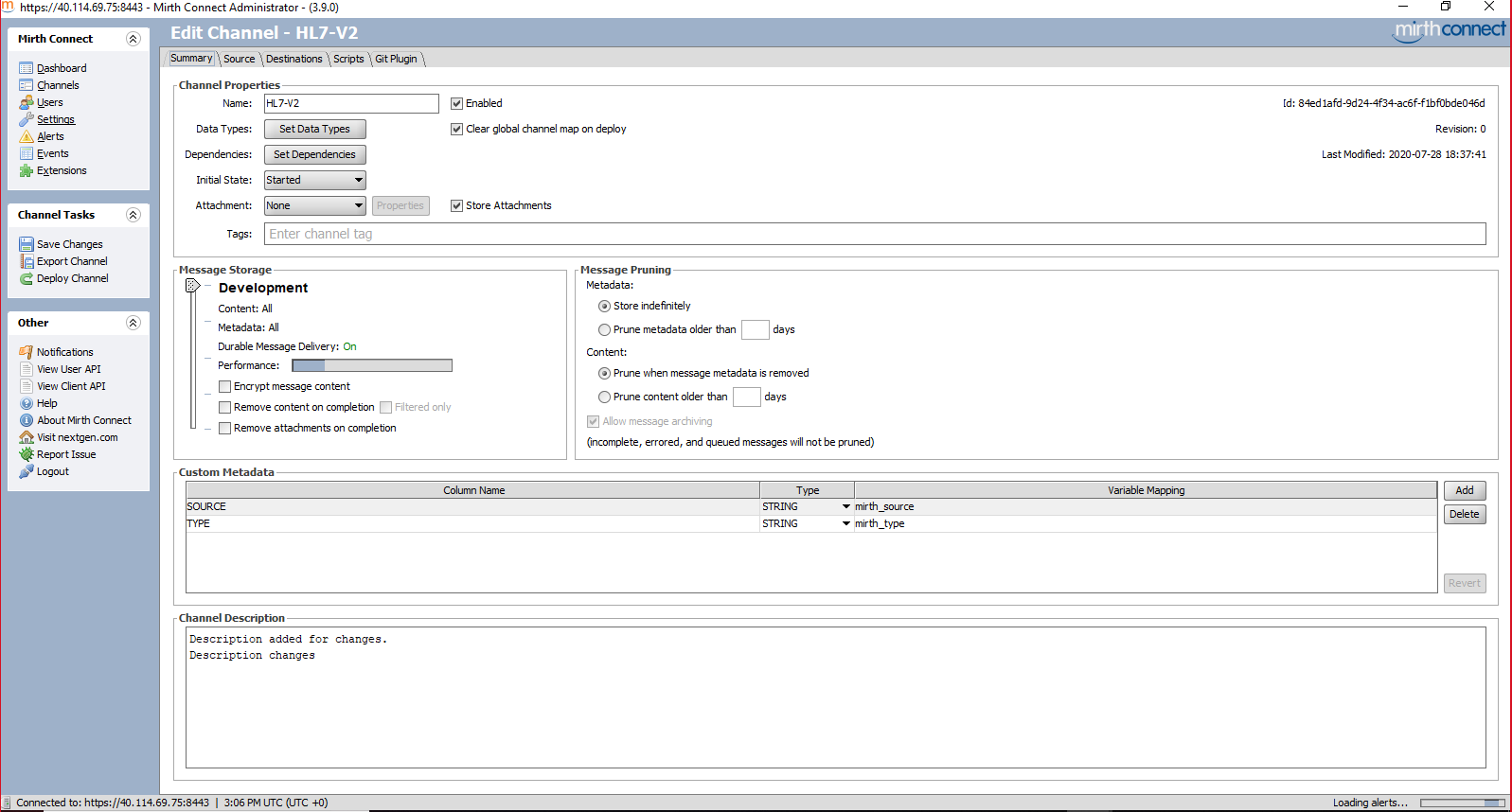


**‘Revert’ any previous Channel Version:**

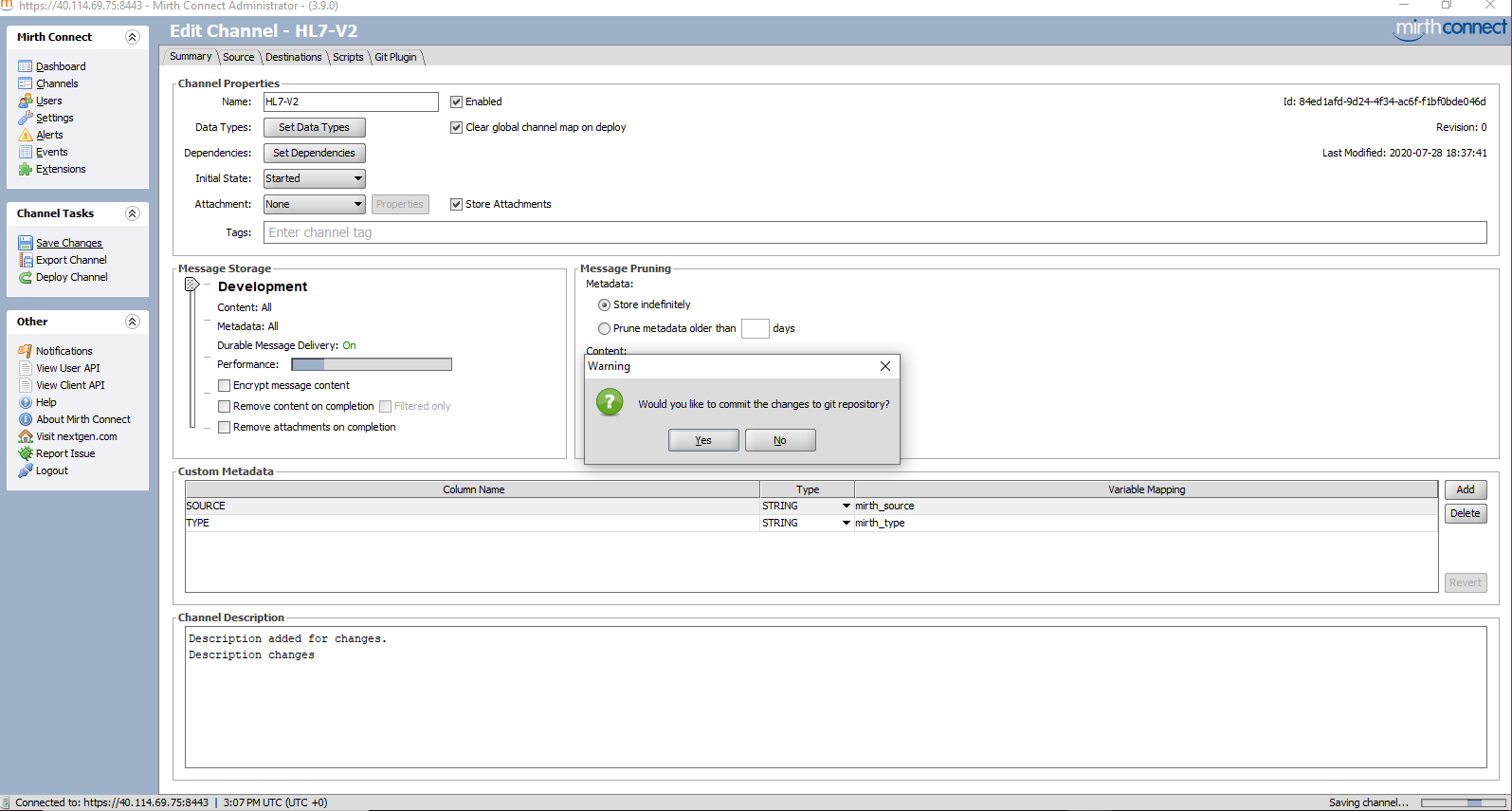
1. To ‘Revert’ any particular previous version of the channel configuration from the available history of the channel configurations, select the check box against any one particular previous version and click on ‘Revert’.



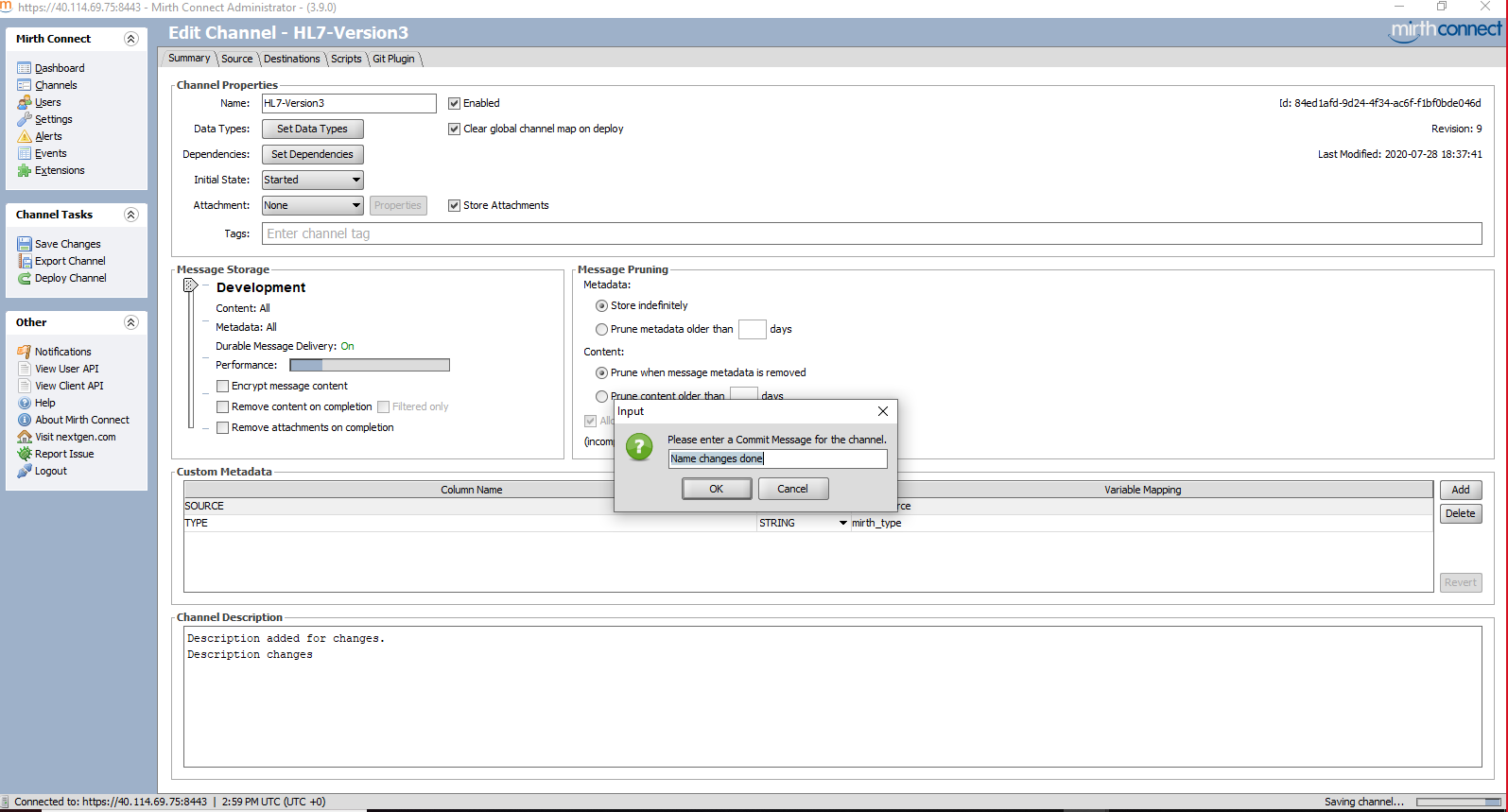
1. Reverting the previous version opens up the channel configurations as per the selected version and navigates user to the ‘Summary’ tab for making any other edits.



1. Once all the changes have been configured select ‘Save Changes’ in Channel Task panel from left Menu Panel.
2. If in the Git Configurations ‘No’ option is selected under ‘Production Environment’ option, then user shall se a warning pop-up asking whether all the changes shall be saved to ‘GIT repository’? Select ‘Yes’ to save the changes to GIT repository or else select ‘No’.



1. After selecting either of the options from above step user is displayed a pop-up to enter the ‘Commit’ comments. Add comments and select ‘OK’ button.



1. After selecting ‘Ok’ button all the configurations are saved and the channel configurations are reverted as per the earlier selected channel version.

