CodeStreamX Studio - User Guide

Contents

Chapter 1. Set Up a Collaborative Project

Overview of CodeStreamX Studio

Here you will learn what CodeStreamX Studio is, focusing on its role and functionalities tailored to collaborative software development projects.

CodeStreamX Studio is an Integrated Development Environment (IDE) designed to streamline the software development process. It offers a comprehensive suite of tools for code writing, editing, debugging, and version control, making it an ideal platform for individual developers and teams alike.

Table 1. CodeStreamX Studio - Basic Features

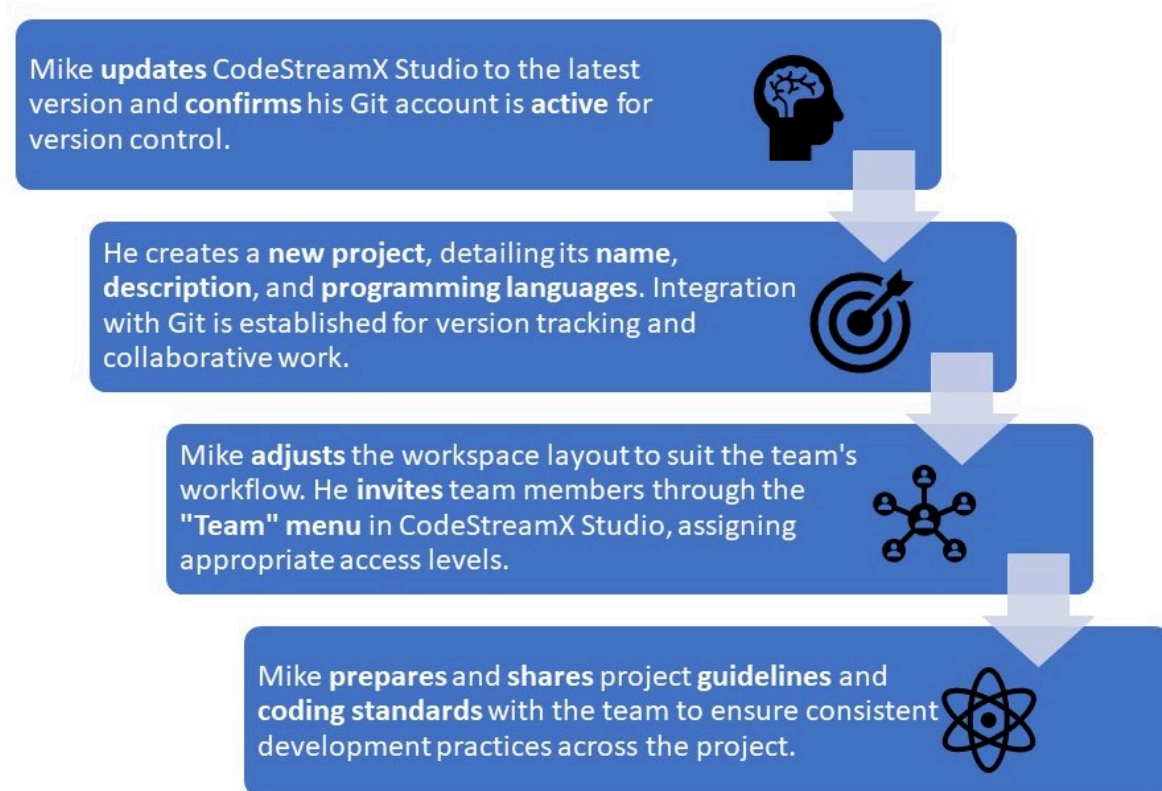| Feature	Description
Version Control Integration	Supports direct integration with version control systems (VCS) like Git, allowing users to manage project versions and collaborate with team members seamlessly within the IDE.
Project Management	Facilitates the creation, organization, and management of software development projects. Users can set up new projects, define project details, and customize the development environment to suit their needs.
Collaborative Tools	Includes features for team collaboration, such as the ability to invite team members to projects, assign roles and access levels, and share project documentation and guidelines.
Customizable Workspace	Offers a flexible workspace that can be tailored to individual preferences and project requirements, enhancing productivity and efficiency.

How to Set Up a Collaborative Project in CodeStreamX Studio

Here you will learn how to set up a new collaborative project in CodeStreamX Studio, including initializing a repository and inviting team members.

To set up a collaborative project, first ensure CodeStreamX Studio is installed and updated on your computer, and have an active account with a compatible version control system like Git. Familiarize yourself with basic version control operations and prepare a new software development project for collaboration in CodeStreamX Studio. Identify team members for collaboration, ensuring all have necessary access and a stable internet connection for project setup and management.

1. Launch CodeStreamX Studio and select "New Project" from the "File" menu.
2. Enter your project details (name, description, programming language).
3. Initialize a new version control repository by selecting "VCS Integration" and choosing your platform (e.g., Git, SVN).
4. Set up the project workspace by customizing the layout and tools according to your team's needs.
5. Invite team members by navigating to the "Team" menu and selecting "Invite Members". Enter their email addresses, and customize their access levels.
6. Share project guidelines and coding standards through the "Project Documentation" section.

Following the steps outlined ensures the successful setup of a collaborative project in CodeStreamX Studio, enabling efficient team-based development and version control integration.

Mike **updates** CodeStreamX Studio to the latest version and **confirms** his Git account is **active** for version control.

He creates a **new project**, detailing its **name**, **description**, and **programming languages**. Integration with Git is established for version tracking and collaborative work.

Mike **adjusts** the workspace layout to suit the team's workflow. He **invites** team members through the **"Team"** menu in CodeStreamX Studio, assigning appropriate access levels.

Mike **prepares** and **shares** project **guidelines** and **coding standards** with the team to ensure consistent development practices across the project.

Chapter 2. Customize Your Workspace

What is Workplace Customization in CodeStreamX Studio?

Here you will find out what workplace customization is and what it includes in CodeStreamX Studio.

Workspace customization in CodeStreamX Studio is a process which involves tailoring the IDE settings and layout to match user preferences and needs. It covers adjusting panels like the code editor, file explorer, and console output for better access, choosing visual themes to ease eye strain and enhance readability, and setting up keyboard shortcuts for a more efficient workflow.

How to Customize Your Workspace in CodeStreamX Studio

Here you will learn the procedure for configuring your workspace in CodeStreamX Studio, aimed at optimizing productivity and aligning the environment with individual development preferences.

Before customizing your workspace in CodeStreamX Studio, ensure you have the latest version installed and are familiar with its interface. Identify your specific development preferences, such as layout, theme, and keyboard shortcuts, to streamline the customization process.

1. Launch CodeStreamX Studio and open the "Preferences" menu from the main toolbar.
2. Layout Customization: Navigate to "View > Layout" to adjust the layout. You can drag and drop different panels (e.g., code editor, file explorer, and output console) to reposition them within the workspace.
3. Theme Selection: Within the "Preferences" menu, select "Appearance". Here, you can choose from a variety of themes to change the color scheme of your IDE, reducing eye strain and improving code readability.
4. Setting Up Keyboard Shortcuts: To further streamline your workflow, customize keyboard shortcuts tailored to your development needs. Go to "Preferences > Keyboard Shortcuts" and refer to the list of default shortcuts. Here, you can assign new shortcuts or modify existing ones to better suit your coding habits.
5. Save Your Customizations: After making your adjustments, click "Apply" to save your customized settings, then "OK" to exit the Preferences menu.

   To enhance your efficiency, customize keyboard shortcuts according to your development needs. Go to "Preferences > Keyboard Shortcuts" and review the list of default shortcuts. Here, you can assign new shortcuts or modify existing ones to align with your coding habits. For a comprehensive list of keyboard shortcuts and tips on customization, refer to CodeStreamX Studio Keyboard Shortcuts

Following the above-mentioned steps results in a personalized and optimized CodeStreamX Studio workspace that aligns with individual development preferences, enhancing productivity and coding efficiency.

CodeStreamX Studio Keyboard Shortcuts

Here you will find a list of keyboard shortcuts in CodeStreamX Studio.

Table 2. File Operations

Operation	Shortcut
New Project	Ctrl + N
Open Project	Ctrl + O
Save	Ctrl + S
Close Project	Ctrl + W

Table 3. Editing Commands

Operation	Shortcut
Cut	Ctrl + X
Copy	Ctrl + C
Paste	Ctrl + V
Undo	Ctrl + Z
Redo	Ctrl + Y

Table 4. Navigation and Search

Operation	Shortcut
Find	Ctrl + F
Replace	Ctrl + H
Go to line	Ctrl + G

Table 5. Debugging and Testing

Run	F5
Debug	F9
Step over	F10
Step into	F11

Chapter 3. Manage Project Versions

What are Project Versions in CodeStreamX Studio?

Here you will learn what project versions are in CodeStreamX Studio.

Project versions are distinct iterations of a software project, captured and managed within a version control system. They document the project's evolution over time, enabling developers to track progress, manage updates, and facilitate collaboration by preserving the state of the project at various development milestones.
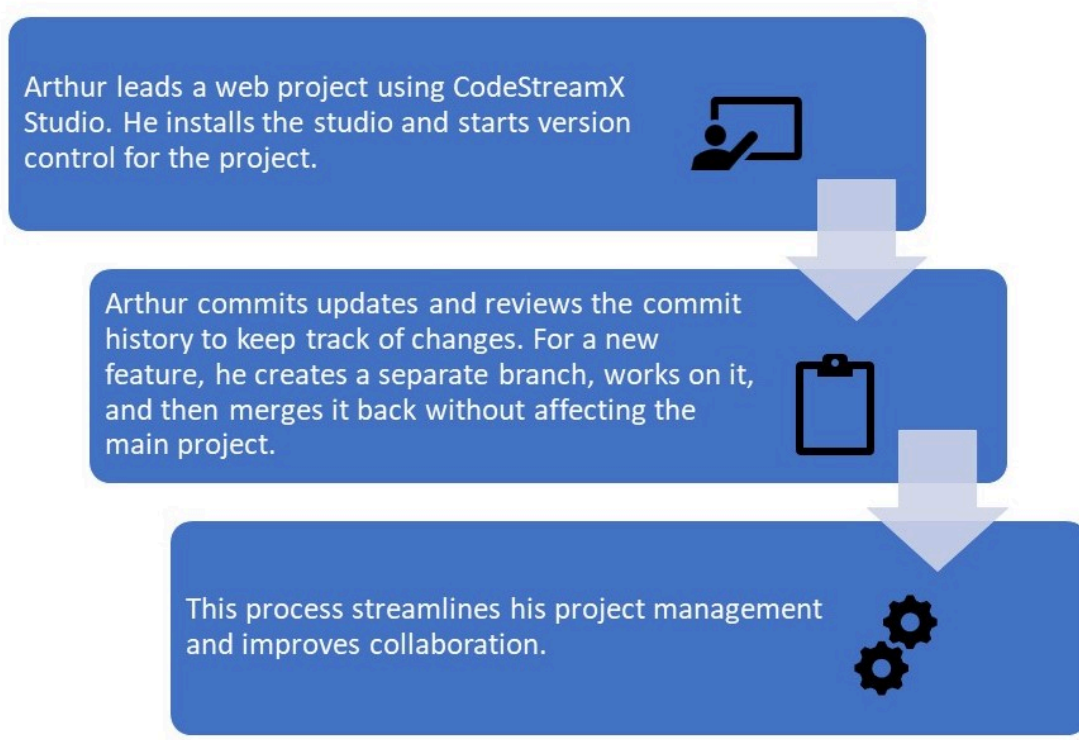
How to Manage Project Versions with CodeStreamX Studio

Learn how to manage project versions in CodeStreamX Studio using its integrated version control, including setup, committing changes, and tracking history. This guide simplifies collaboration and version tracking within the IDE, enhancing project organization and development efficiency.

Before managing project versions with CodeStreamX Studio, ensure the IDE is installed and you understand its Git integration. Basic knowledge of version control operations is essential. Have a project for version management and ensure a stable internet for remote repository access.

1. Initialize Version Control: Launch CodeStreamX Studio, load your project, and select "Initialize Repository" from the "VCS" menu to start tracking your project with version control.
2. Commit Changes: After updating your project, navigate to "VCS > Commit Changes," enter a summary of the changes in the commit message, and click "Commit" to save the new state of your project.
3. Review Commit History: To inspect the record of changes, go to "VCS > Show History." This displays a log of all commits, showing the modifications, the contributors, and the timing for each.
4. Branch for Feature Development: Use the version control system to create new branches. This method enables isolated development of features or fixes, keeping the main project stable.
   For detailed version control commands in CodeStreamX Studio, see Version Control Commands in CodeStreamX Studio .

Executing these steps results in streamlined project version control within CodeStreamX Studio, enabling precise change tracking, efficient progress documentation, and isolated feature development through branching, thereby optimizing collaboration and project management.

Arthur leads a web project using CodeStreamX Studio. He installs the studio and starts version control for the project.

Arthur commits updates and reviews the commit history to keep track of changes. For a new feature, he creates a separate branch, works on it, and then merges it back without affecting the main project.

This process streamlines his project management and improves collaboration.

Version Control Commands in CodeStreamX Studio

Here you will find version control commands in CodeStreamX Studio.

Table 6.

Command	Description
Initialize Repository	Prepares the current project for version tracking by creating a new repository.
Commit Changes	Saves the current state of the project as a new version in the repository's history.
Show History	Displays a chronological list of all commits made to the project, including details about each commit.
Create Branch	Creates a new branch in the repository, allowing for parallel development streams.

Table 6. (continued)

Command	Description
Merge Branch	Combines changes from one branch into another, typically used to integrate feature development back into the main project.
Fetch from Remote	Updates the local repository with changes made in the remote repository without merging.
Pull from Remote	Fetches changes from the remote repository and immediately merges them into the local repository.
Push to Remote	Uploads local repository changes to a remote repository to share with others.