Learn How to Comment Blocks of Code in VBA with Examples

Authored by

Mohammed looti

November 15, 2025

In the professional realm of software development, particularly when working within specialized environments like Visual Basic for Applications (**VBA**), the disciplined practice of effective code commenting is far more than a professional courtesy--it is a foundational requirement for successful, maintainable project execution. High-quality comments significantly boost code readability, drastically improve long-term maintainability, and are absolutely essential for any collaborative development efforts. In **VBA**, the established syntax for commenting a single line is the single quotation mark ('). This character acts as a compiler delimiter, instructing the program to ignore the remainder of the line, making it perfect for quick inline explanations or temporarily disabling a single statement.

However, **VBA** developers frequently encounter a significant roadblock to productivity: the default absence of a streamlined, built-in feature for commenting out large, continuous blocks of code simultaneously. While it is technically possible to manually insert an apostrophe at the start of dozens or even hundreds of lines, this highly repetitive task quickly becomes cumbersome and severely inefficient. This fundamental limitation greatly hampers the developer workflow, especially during intensive debugging sessions, where complex code sections must be rapidly toggled on and off for isolation testing, or when experimenting with alternative logical implementations. Relying on this manual process distracts the developer from core problem-solving tasks and introduces unnecessary friction into the crucial development cycle.

Fortunately, the VB Editor, which serves as the primary integrated development environment for **VBA**, is designed with powerful customization capabilities. These advanced options allow users to effectively bridge this functional gap by personally tailoring the environment to their specific needs. By leveraging these advanced configuration settings, it is entirely possible to create and assign a dedicated keyboard shortcut that enables the swift commenting and uncommenting of entire code blocks with a single, efficient keystroke. This transformative enhancement dramatically streamlines the coding process, making it far more efficient and user-friendly. The following comprehensive, step-by-step guide details the precise process required to activate and configure this invaluable feature, moving you toward a significantly more productive **VBA** development experience.

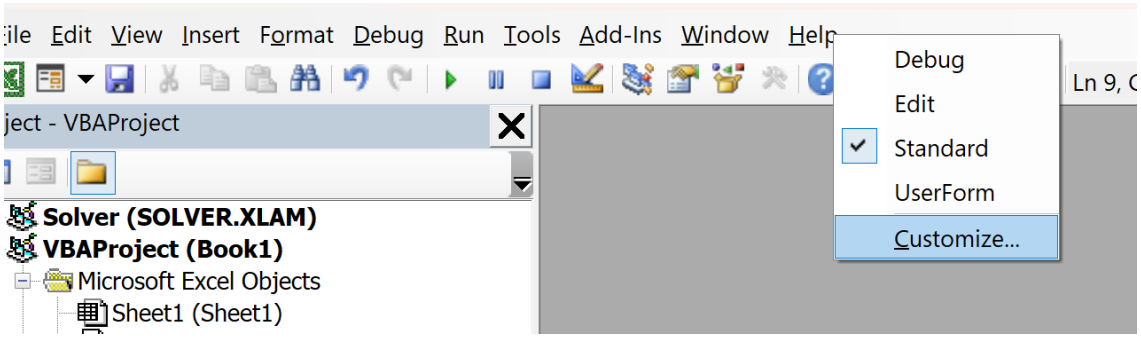
Step 1: Accessing the VB Editor Customization Interface

The first foundational step toward achieving highly efficient code management begins directly within the **VB Editor** (VBE) itself. This environment is the dedicated workspace where all **VBA** modules, macros, and user forms are meticulously crafted, debugged, and maintained. Recognizing the diverse and evolving needs of professional developers, Microsoft intentionally designed the **VB Editor** to be exceptionally flexible, offering deep customization options that allow users to tailor the user interface and workflow to meet specific project demands and personal preferences. The crucial initial step involves locating and accessing the main customization menu, which acts as the gateway to integrating new tools and features into the editor's standard

appearance and functional set.

To initiate this vital customization process, you must navigate directly to the primary toolbar located at the very top of the **VBA** Editor window. Specifically, identify any empty or non-icon space on this toolbar area. Performing a right-click action on this empty area will immediately invoke a context-sensitive menu. Among the various options presented in this important menu, the one we are interested in is **Customize**. Selecting this option will open the comprehensive "Customize" dialog box, which serves as the central control panel for all aesthetic and functional modifications within the **VB Editor** environment.

This initial preparatory action is foundational; it prepares the entire development environment for the subsequent integration of the specialized block commenting tools. The "Customize" dialog box is logically structured, typically featuring several tabs and hierarchical categories that organize the vast array of available commands, existing toolbar buttons, menu options, and keyboard preferences. Understanding how to correctly access and navigate this panel is absolutely essential for personalizing your **VBA** coding experience, ensuring that frequently used functions are always within immediate reach. This customization process significantly helps to unlock the full potential for development efficiency.



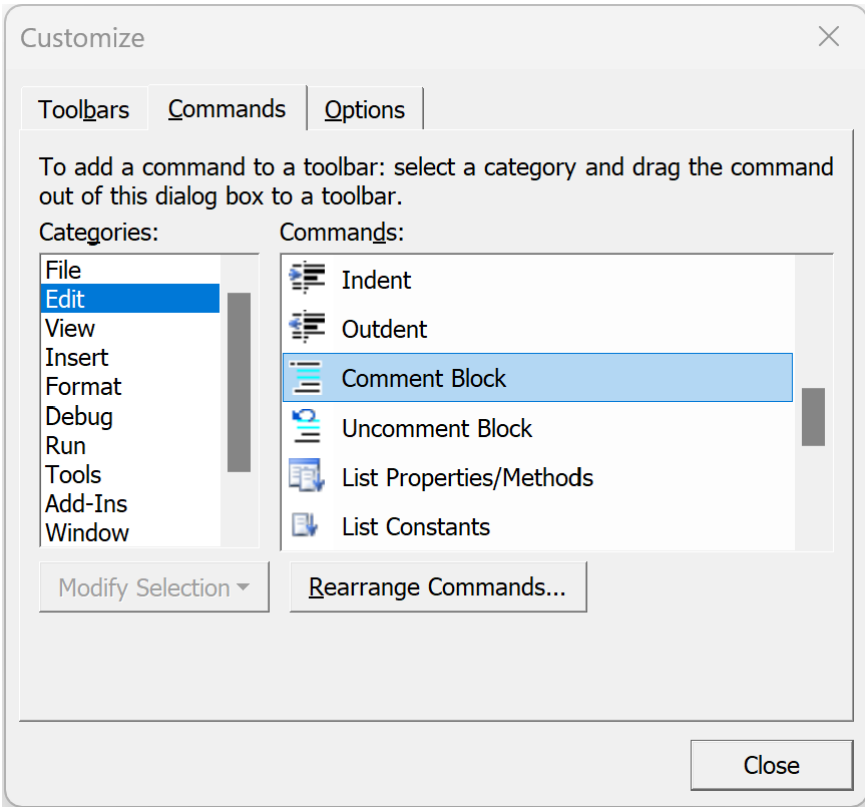
Step 2: Integrating the Comment Block Command into the Toolbar

Once the "Customize" dialog box has been successfully launched and is visible on your screen, the immediate objective shifts to locating and transferring the specialized "Comment Block" command onto your active **VBA** toolbar. It is important to realize that this crucial command is typically hidden from the default interface view, yet it remains fully accessible within the extensive catalog of functionalities that the **VB Editor** is capable of exposing. The "Customize" interface is meticulously organized to facilitate rapid command discovery and integration.

To begin this integration, click firmly on the **Commands** tab, which is logically positioned within the "Customize" dialog box. This tab systematically lists every command available for integration, often categorized to simplify navigation and search. Next, direct your attention to the *Categories* list

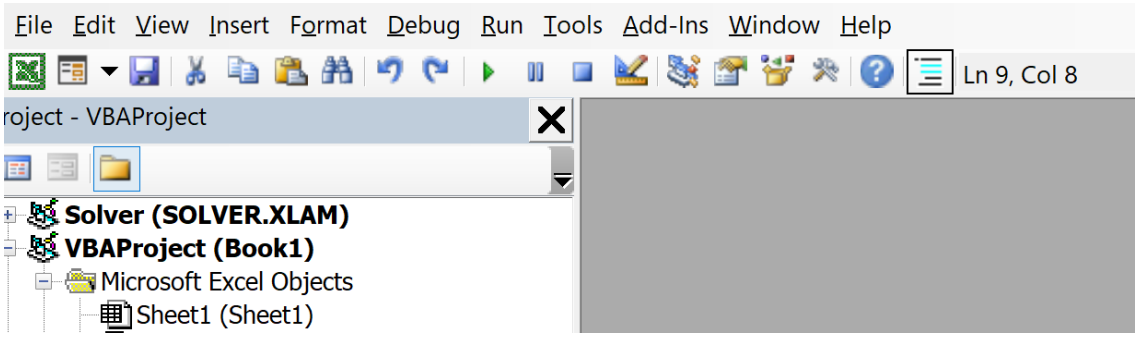
situated on the left-hand side of the dialog window. Scroll down through this list until you find, and then confidently select, the **Edit** category. The "Edit" grouping contains a focused collection of commands that are primarily utilized for advanced text manipulation, comprehensive code formatting, and, most importantly for the purposes of this guide, specific code structure modification tasks.

After successfully highlighting the **Edit** category, the corresponding *Commands* list on the right will instantly populate with all the related editing functions. Within this list, you must identify and select the command explicitly labeled **Comment Block**. With this command active, the final, intuitive step involves employing a simple drag-and-drop action: click and hold the **Comment Block** command and physically drag it directly onto your chosen location on the active **VBA** Editor toolbar. As you move the command, a helpful visual indicator will appear, guiding you to a suitable placement. Releasing the mouse button completes the process, and a distinct, new icon representing the "Comment Block" functionality will instantly appear on your toolbar, confirming its successful integration into your workspace.



The immediate appearance of this specialized icon on your **VBA** Editor toolbar serves as a clear visual confirmation of the successful customization effort. This newly available feature significantly enhances your immediate workflow, already eliminating the tedious, manual requirement of inserting a single quotation mark on multiple lines. While the button is fully functional at this stage,

reliance solely on repetitive mouse clicks can still interrupt the critical flow state during coding. Therefore, the subsequent and most important step will focus on assigning a highly efficient keyboard shortcut to this button, transforming it from a mere clickable icon into an exceptionally powerful tool for rapid, seamless code manipulation. This seamless integration is a critical stride toward establishing a truly customized and highly productive development environment.

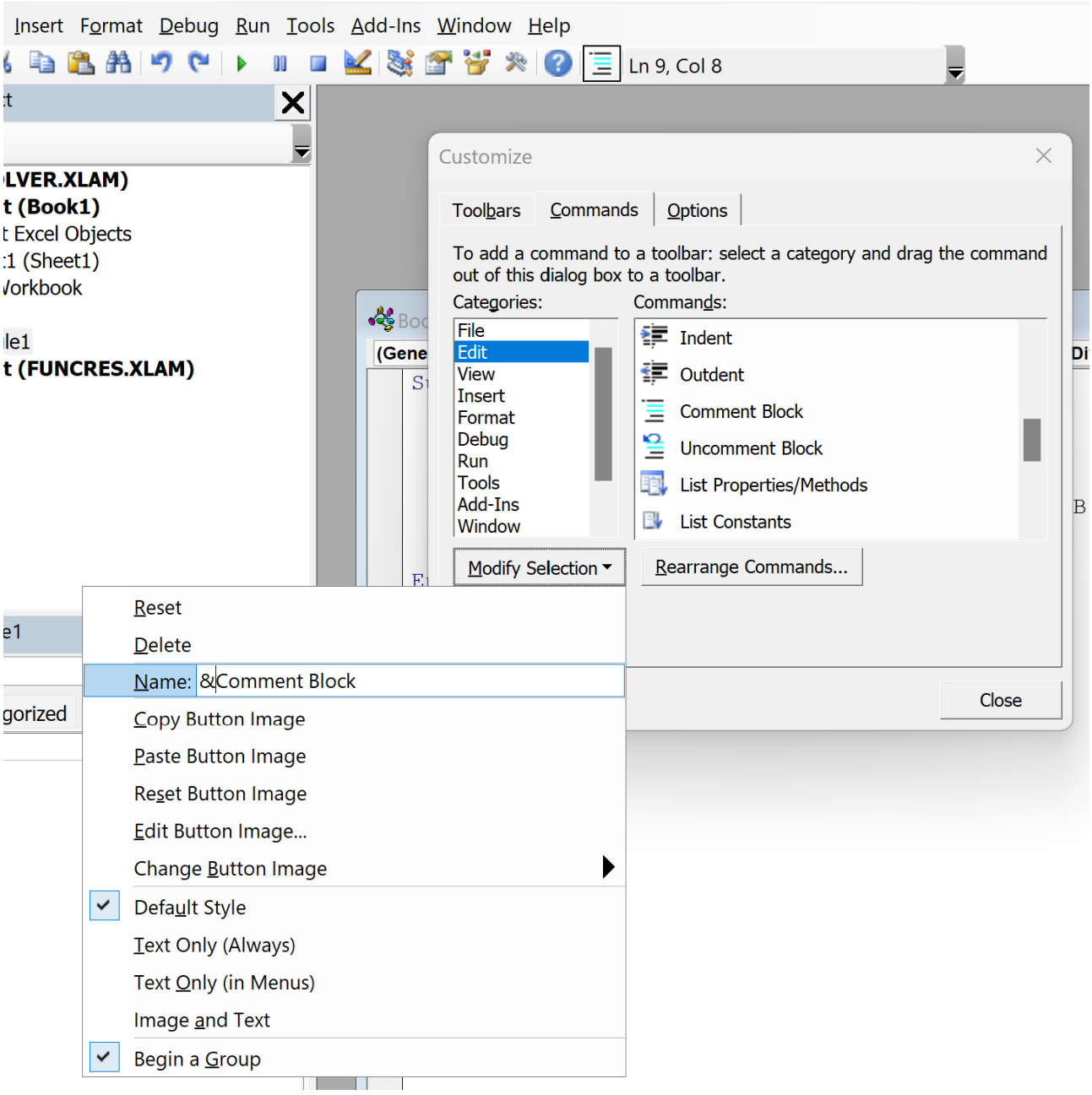


Step 3: Crafting a Keyboard Shortcut for Enhanced Efficiency

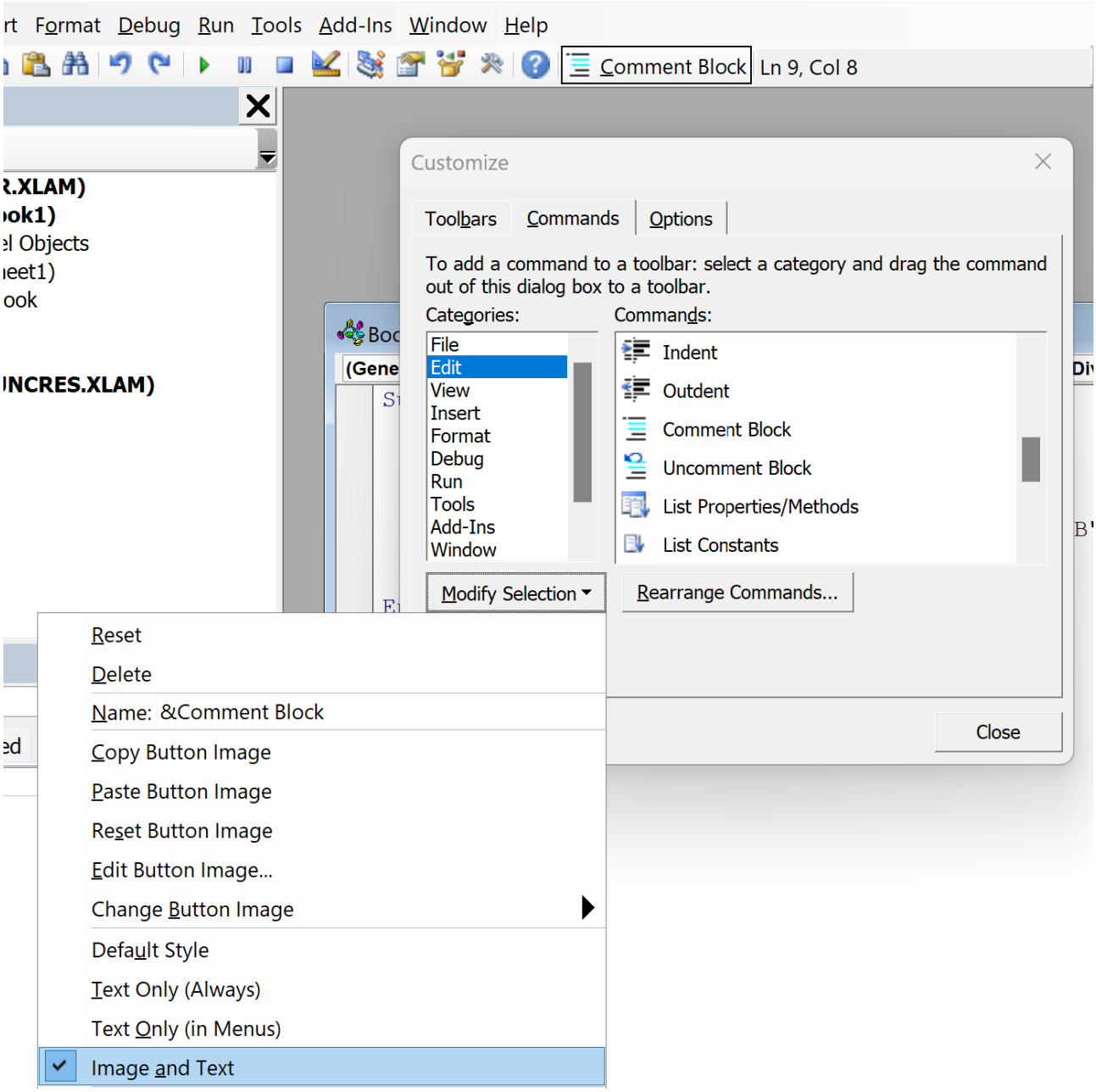
The physical addition of the **Comment Block** button to the toolbar marks a significant improvement, but to achieve maximum coding speed and efficiency, the feature must be immediately accessible without requiring the developer to move their hands away from the keyboard. Assigning a dedicated accelerator key, universally known as a keyboard shortcut, is the definitive method for achieving unparalleled speed and maintaining the flow state during rigorous coding sessions. This key combination allows for instantaneous execution of the function.

To commence the shortcut creation process, you must first ensure the newly added **Comment Block** icon is actively selected within the context of the open "Customize" panel. Click directly on the icon on the toolbar; a subtle black border will appear around it, unequivocally confirming its selection status. Once the button is selected, redirect your focus to the settings visible within the "Customize" dialog box.

Within the dialog, locate the **Modify Selection** dropdown menu. Clicking this menu will reveal a set of options governing how the selected element can be altered. Choose the **Name** field option. This is where the mechanism of the accelerator key is implemented. You must strategically insert an ampersand symbol (**&**) immediately preceding the specific character you wish to designate as the shortcut key. For instance, if the current name is "Comment Block," you should modify it to "**&Comment Block**". This crucial modification designates 'C' as the key. After typing the ampersand, press the **Enter** key to finalize and confirm the change. The ampersand convention designates the following character--in this case, 'C'--as the accelerator key, meaning that the command will now be instantly activated by pressing the **Alt + C** key combination. This is now your powerful, rapid-fire shortcut for commenting blocks of **VBA** code.



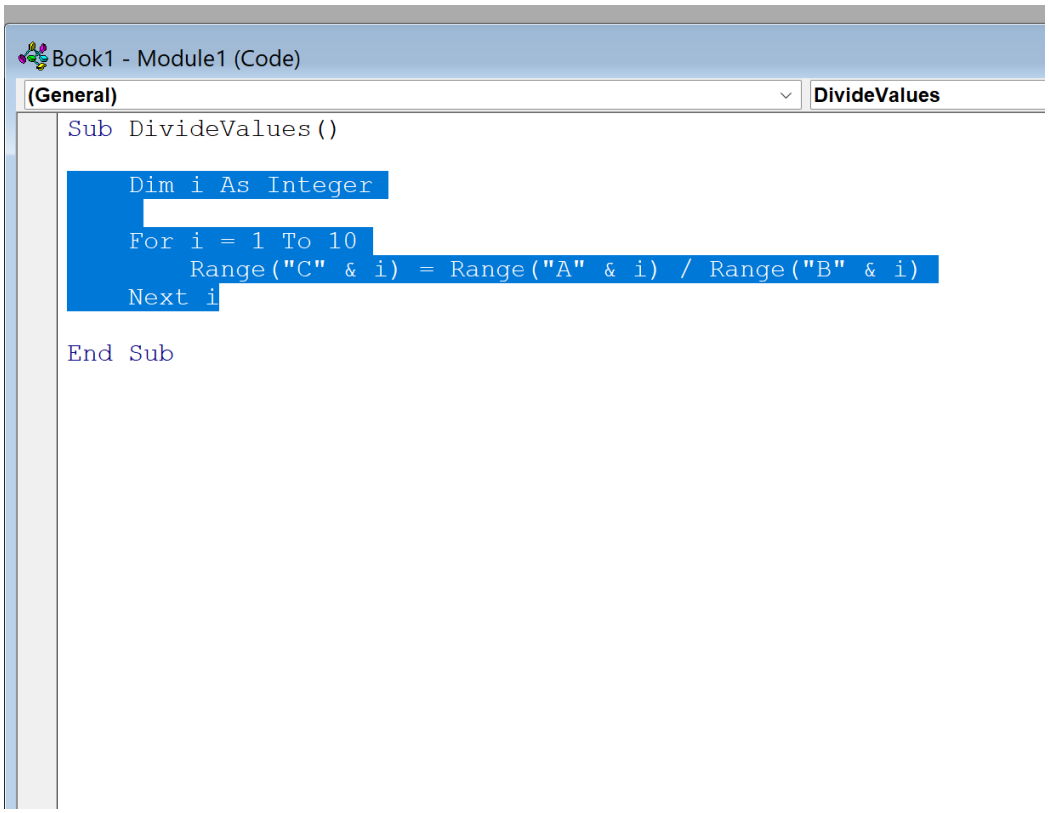
Immediately after confirming the accelerator key assignment, you might observe a visual change in the button's appearance on the toolbar; often, the icon might temporarily disappear, leaving only the text. To ensure optimal usability and visual clarity, you must adjust the display settings one final time. Click the **Modify Selection** dropdown once more. From the list of available presentation options, select **Image and Text**. This choice ensures that your toolbar button displays both its original illustrative icon and the descriptive text "Comment Block" simultaneously. Crucially, this visual optimization is achieved without compromising the functionality of your newly established **Alt + C** keyboard shortcut. This completes the comprehensive setup for the block commenting feature, effectively transforming a multi-step, manual process into an efficient, instantaneous keyboard command, dramatically boosting your development speed and coding agility.



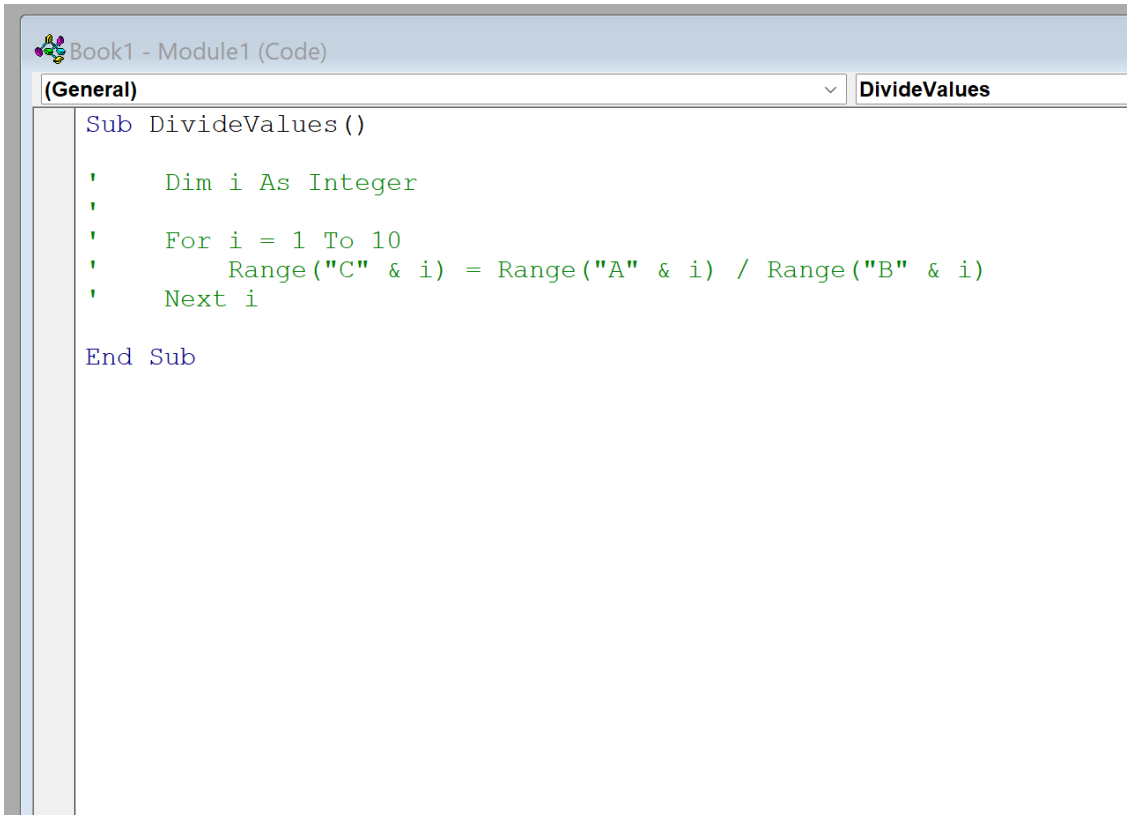
Step 4: Practical Application of the Comment Block Shortcut

With the **Comment Block** shortcut now fully configured and accessible via the highly efficient **Alt + C** combination, you are fully prepared to realize the substantial increase in your **VBA** development productivity. This feature proves invaluable in several common development scenarios, such as temporarily isolating and disabling specific code sections for targeted unit testing, significantly simplifying the debugging process, or when iterating on different structural solutions without needing to permanently delete earlier, valid logic. The operational use of this new feature is intuitive and aligns perfectly with the expected behavior of productivity-enhancing keyboard commands across all modern applications.

Consider a real-world programming scenario: you are working on an extensive sub procedure or a complex function containing numerous lines of executable code that must be temporarily deactivated for testing a new dependency. Traditionally, this required tediously inserting a single quote mark at the beginning of every line--a task that is not only time-consuming but highly susceptible to human error. The new shortcut provides an elegant and immediate solution. The image below illustrates a typical sub procedure containing operational code that we will use to demonstrate the power and efficiency of the new block commenting feature.



To properly utilize the shortcut, the first critical step is to accurately highlight the entire contiguous block of code that you intend to comment out. This selection can be efficiently executed by clicking and dragging the cursor over the relevant lines, or by using precise keyboard navigation techniques such as **Shift + Arrow keys** for maximum precision. Once the desired segment of code is fully selected, immediately execute the assigned shortcut key combination: **Alt + C**. The effect is instantaneous: every single line within the highlighted block will be immediately prefixed with the single quotation mark, successfully rendering the entire section inert and commented out. This transformation is immediate, clearly visible in the code window, and significantly enhances the efficiency of your code management during active development and iterative testing phases.



Step 5: Implementing the Complementary Uncomment Block Shortcut

While the ability to quickly comment out code blocks is a primary accelerator for development and testing, the capability to rapidly revert this change--to uncomment the code--is equally vital for maintaining a seamless and non-disruptive workflow. After successfully testing or modifying a temporarily disabled section, you must reactivate it without the laborious necessity of manually removing each individual single quote mark. Fortunately, the **VBA** Editor offers a perfectly symmetrical and complementary feature: the "Uncomment Block" command. This feature can and should be assigned its own dedicated keyboard shortcut, mirroring the setup procedure used for the "Comment Block" functionality, ensuring complete and immediate control over code activation and deactivation.

To configure the "Uncomment Block" shortcut, you will repeat the exact sequence of steps detailed in the preceding sections. Begin by accessing the "Customize" dialog box once more (right-click on an empty space on the toolbar and select **Customize**). Navigate to the **Commands** tab, select the **Edit** category from the left pane, and then meticulously locate **Uncomment Block** within the *Commands* list on the right. Once found, drag this command option directly onto your active toolbar, ideally positioning it adjacent to the "Comment Block" icon for logical grouping and ease of use.

With the "Uncomment Block" icon successfully placed on your toolbar, click it to select it within the "Customize" dialog box. Next, utilize the **Modify Selection** dropdown menu, and within the **Name** field, strategically place an ampersand (**&**) before the 'U' in "Uncomment Block" (e.g., "**&Uncomment Block**"). Press **Enter** to confirm this change. This action assigns **Alt + U** as the designated accelerator key for the uncommenting process. Finally, return to the **Modify Selection** menu and ensure the display option is set to **Image and Text** for maximum clarity. This powerful, symmetrical pair of shortcuts--**Alt + C** for commenting and **Alt + U** for uncommenting--grants the developer precise and immediate control over code blocks, dramatically enhancing both the speed of development and the overall quality of code management within the **VBA** environment.

Additional Resources for Advanced VBA Development

Mastering the efficient management of code blocks through customizable shortcuts is an important milestone in becoming a proficient **VBA** developer. The ability to rapidly navigate, toggle, and manipulate code is absolutely crucial for building solutions that are not only robust but also scalable and easy to maintain over time. To further solidify your expertise in **VBA** and equip yourself to handle a wider spectrum of complex programming tasks, we highly recommend exploring the following tutorials and specialized resources. These guides delve into other essential functionalities, best practices, and foundational concepts that will empower you to consistently write cleaner, more effective, and professional-grade **VBA** code.