SYMBOLA

VBCC compiler setup on AmigaOS

Table of Contents

Disclaimer

This guide is provided for informational purposes only. Every effort has been made to ensure the accuracy and completeness of this guide, but it is provided “as is” without warranty of any kind, express or implied. The author shall not be held liable for any direct, indirect, incidental, or consequential damages or losses arising from the use of this guide.

The procedures and software described in this guide are subject to change and may not be up-to-date. Users are advised to exercise caution and consider their specific circumstances when following the instructions.

This guide may contain links to external websites. The author is not responsible for the content or accuracy of any external site.

Please use this guide responsibly and at your own risk.

Contact the Author

If you've spotted an error or simply wish to make contact, feel free to leave a message at:

https://symbola.co.uk/contact/

Your feedback and inquiries are always welcome!

Preface

Welcome to the guide on setting up VBCC for AmigaOS 3.x. This document is designed to help enthusiasts and developers alike in installing and using the VBCC compiler on their Amiga systems, specifically focusing on the AmigaOS 3.x environment. With a nod to the enduring legacy of the Amiga platform, this guide also acknowledges the practicality of using modern emulation tools like Amiga Forever for those who may not have access to physical hardware.

VBCC serves as a bridge between the classic Amiga development experience and modern programming efficiencies. By concentrating on AmigaOS 3.x, we cater to a widely appreciated version of the operating system, ensuring that the instructions remain relevant for a significant portion of the Amiga community.

Throughout this guide, we will navigate the process of downloading, installing, and configuring VBCC, making sure you have all the necessary tools at your disposal. Whether you are looking to revive old projects or embark on new ones, this guide aims to provide a straightforward path to achieving your development goals on the Amiga platform.

Hardware / Software Requirements

Before diving into the setup of VBCC for AmigaOS 3.x, it's essential to ensure you have the right hardware or software environment prepared. This section outlines the necessary requirements to get started.

Real Amiga Hardware: If you're fortunate enough to own an Amiga computer, it's the ideal platform for experiencing true Amiga development. This guide assumes you have an Amiga model capable of running AmigaOS 3.x. Additionally, ensure that your system has sufficient memory and storage to accommodate your development projects and the VBCC compiler itself.

Emulation Setup: For those who don't have access to physical Amiga hardware, emulation is a fantastic alternative that replicates the Amiga experience on modern computers. The most recommended emulator for this purpose is Amiga Forever, which offers a comprehensive package including pre-configured AmigaOS environments, making it an excellent choice for both beginners and seasoned developers. However, other emulators such as WinUAE or FS-UAE are also suitable options, provided they support AmigaOS 3.x environments.

Regardless of the path you choose, the goal is to create a stable and functional AmigaOS 3.x environment where VBCC can be installed and operated efficiently. Ensuring your setup meets these requirements will pave the way for a smooth development experience as you step into the world of Amiga programming.

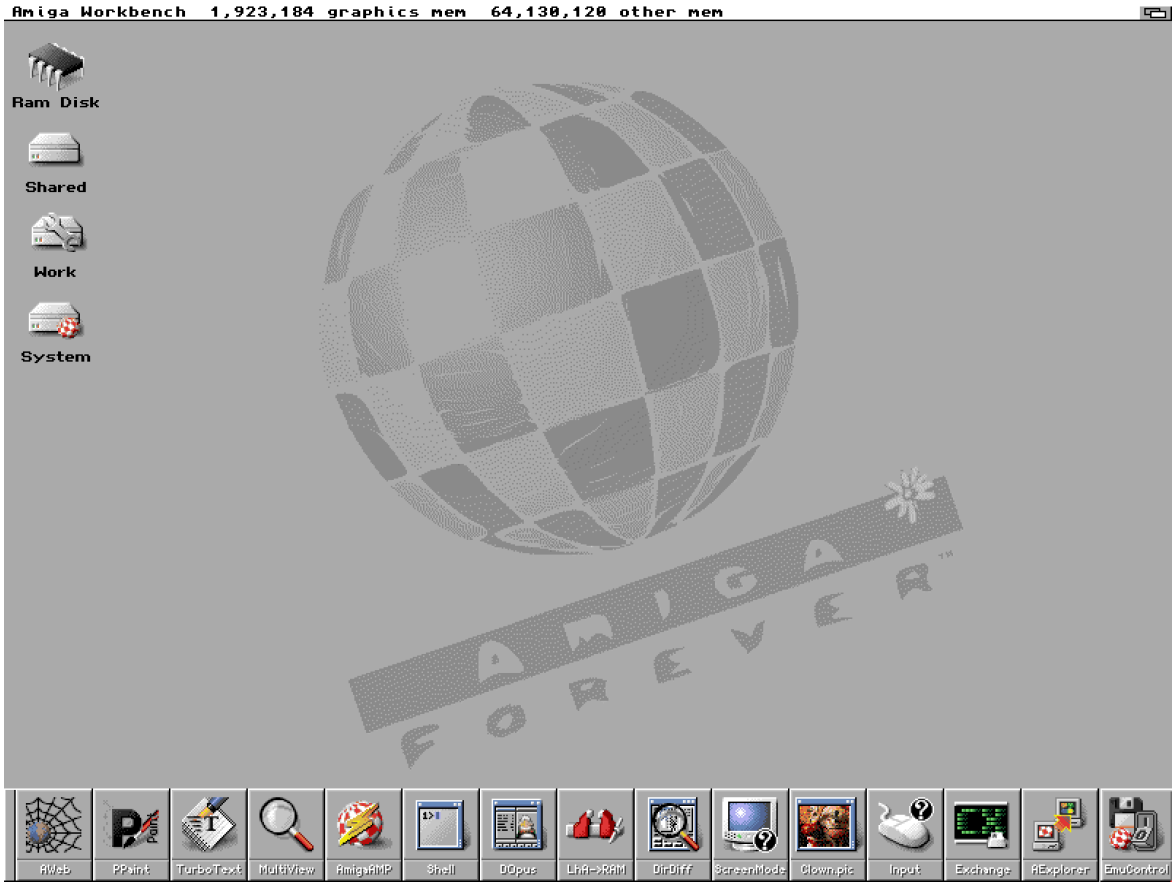
For more information on Amiga Forever, check it out at https://www.amigaforever.com/.

Required Downloads

Before you can start working with VBCC on AmigaOS 3.x, you'll need to download a few essential files. These include the VBCC compiler itself, the VBCC target for Amiga, the NDK 3.9, and the incupd tool. Each of these plays a crucial role in setting up your development environment.

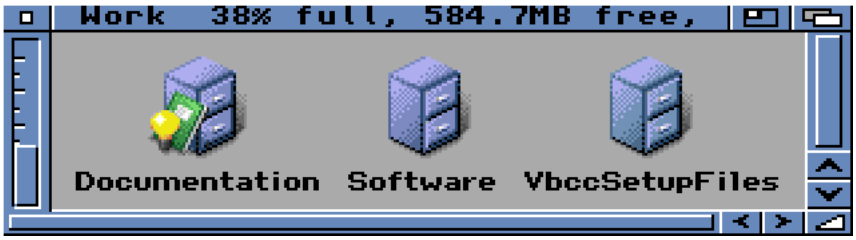
1. **VBCC Compiler:** This is the core component. VBCC is a portable and highly optimizing C compiler that supports multiple target platforms, including Amiga. It's designed to produce efficient code and supports various C standards. http://sun.hasenbraten.de/vbcc/ (**vbcc_bin_amigaos68k.lha**).
2. **VBCC Target for Amiga:** To compile code specifically for AmigaOS, you need the Amiga target files. These include configurations and libraries tailored for Amiga development, ensuring your compiled programs can run on AmigaOS 3.x. http://sun.hasenbraten.de/vbcc/ (**vbcc_target_m68k-amigaos.lha**).
3. **NDK 3.9 (Native Development Kit):** The NDK is an essential set of libraries, headers, and development resources for AmigaOS. Version 3.9 is tailored for AmigaOS 3.x and provides the necessary APIs and documentation for software development. http://hp.alinea-computer.de/AmigaOS/NDK39.lha
4. **incupd (Include Updater):** This tool is used to update the include files in your NDK installation. It ensures that your development environment is up to date with the latest API changes and fixes, making your development process smoother and more reliable. http://aminet.net/package/dev/asm/incupd

Make sure to download each of these files from their respective sources to ensure you have all the necessary components to begin your development journey on AmigaOS 3.x.

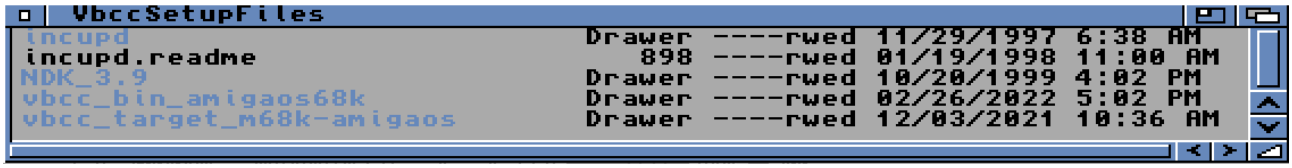


Extracting the Files

After downloading the necessary files for setting up VBCC on your Amiga or emulator, the next step is to extract them to a specific location. This organization will help you manage the components effectively and streamline the setup process.

1. **Create a Folder for VBCC Setup Files:** First, you'll need to create a dedicated folder where you can keep all the downloaded and extracted files. On your Work drive, create a folder named "**VbccSetupFiles**". This will be the central location for all the files you're about to extract.


2. **Extracting the Downloaded Files:** With the folder created, it's time to extract each of the downloaded files into it. You should have the following files:
   - vbcc_bin_amigaos68k.lha
   - vbcc_target_m68k-amigaos.lha
   - NDK39.lha
   - incupd.lha

   Use an extraction tool that supports the "**.lha**" format to extract these files. If you're working on an actual Amiga, tools like LhA can be used. For those on an emulator, you can extract these files on your host system and then transfer them to the emulated Amiga environment.
3. **Place the Extracted Files in "VbccSetupFiles":** Ensure that all extracted files and folders are moved into the "**VbccSetupFiles**" folder you created. This step keeps your workspace organized and makes the next stages of the setup process smoother.


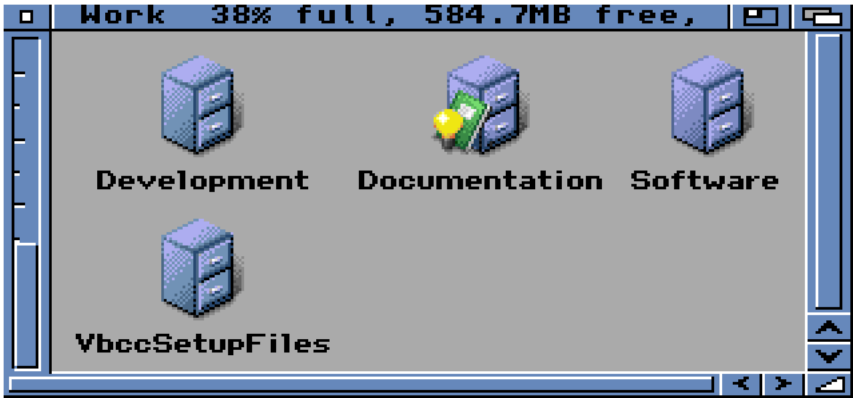
   Extracted setup files shown

By following these instructions, you'll have all the necessary VBCC components ready in one place, setting the stage for the installation and configuration of your Amiga development environment.

Installing and Setting Up VBCC

After preparing your files, the next steps involve copying the necessary components to their correct locations, setting up your environment, and installing VBCC along with its Amiga target. Follow these steps carefully to ensure a smooth installation process.

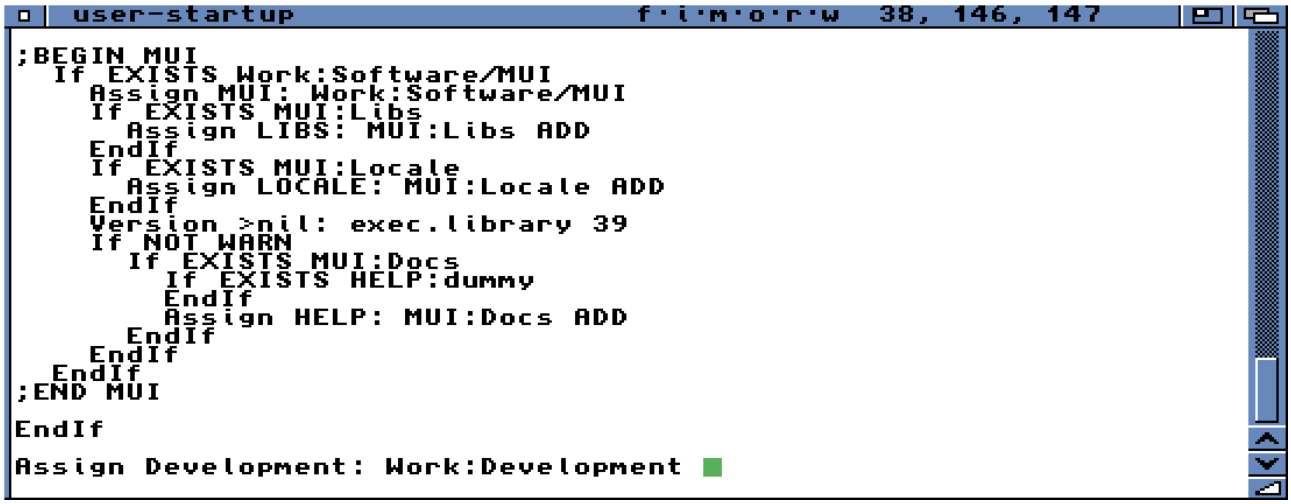
1. **Create a 'Development' Folder:** Begin by creating a new folder named "**Development**" within the "**Work:**" drive. This folder will serve as the primary location for your development tools and libraries.



2. **Copy NDK 3.9 to Your Development Folder:**
   - Navigate to the “**Work:VbccSetupFiles**” folder.
   - Copy the “**NDK_3.9**” folder to “**Work:Development**”.

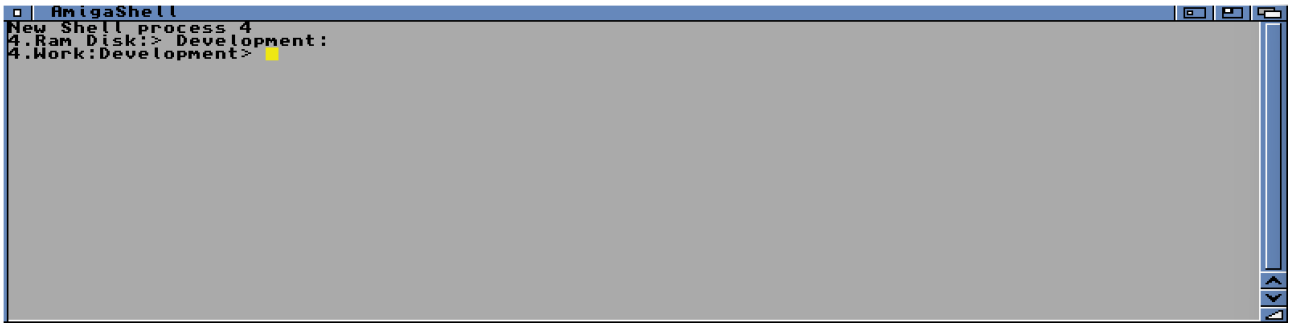
   This places the NDK in a central location for development purposes.
3. **Update the “User-Startup:”**
   - Open the “**S:user-startup**” file located in your system's “**S:**” directory for editing.
   - Add the following line at the end of the file:

     Assign Development: Work:Development
   - Save your changes. This assign makes it easier to access your development folder across the system.



4. **Reboot Your System:**
   - It's important to reboot your Amiga or emulator to ensure that the changes to the “**user-startup**” file are recognized by the system.

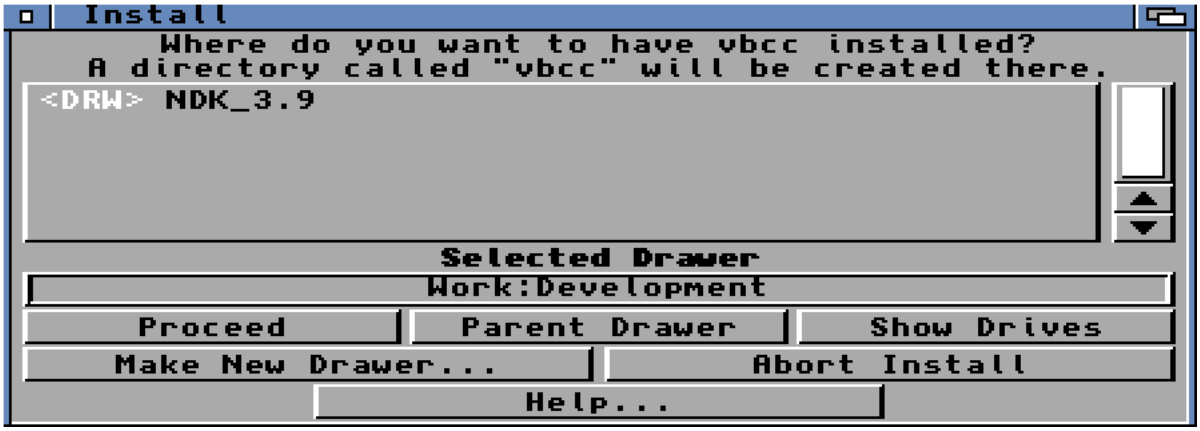
5. **Test the Assign:**
   - After rebooting, open a CLI window and type “**Development:**” to check if the assign was successful. You should see the path change to "**Work:Development**", confirming the assign works as expected.



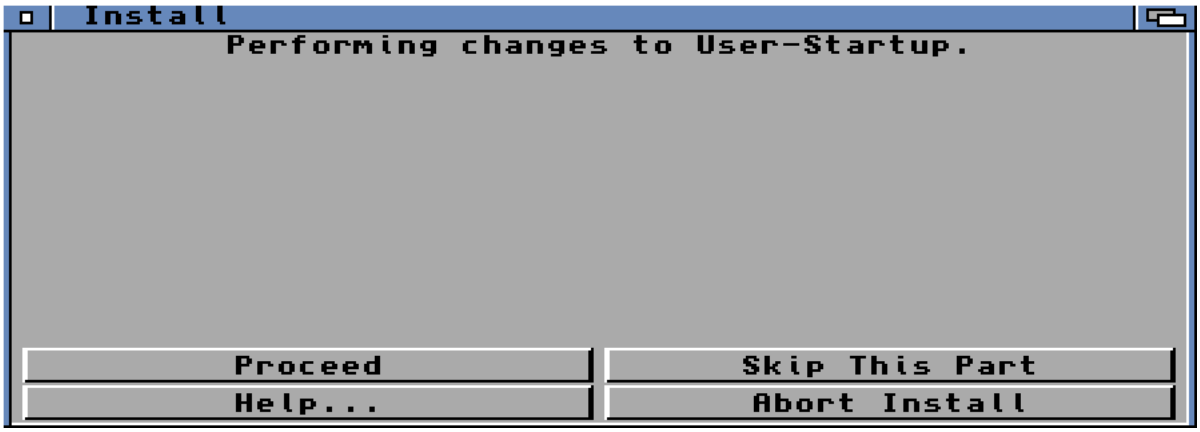
6. **Update Include Files with incupd:**
   - Copy all files from “**Work:VbccSetupFiles/incupd/include/fd**” to “**Development:NDK_3.9/include/fd**”. Overwrite existing files when prompted.
7. **Update dos.h**:
   - Copy “**Work:VbccSetupFiles/incupd/include/dos/dos.h**” to “**Development:NDK_3.9/include/include_i/dos**”.
8. Update execbase.h:
   - Copy “**Work:VbccSetupFiles/incupd/include/exec/execbase.h**” to “**Development:NDK_3.9/include/include_i/exec**”.
9. Update Additional Include Files:
   - Copy all files from “**Work:VbccSetupFiles/incupd/include**” to “**Development:NDK_3.9/include/include_h**”. Overwrite existing files when prompted.

10. **Install VBCC:**
    - Navigate to “**Work:VbccSetupFiles/vbcc_bin_amigaos68k/install**”.
    - Choose "**Expert User**" mode when prompted.

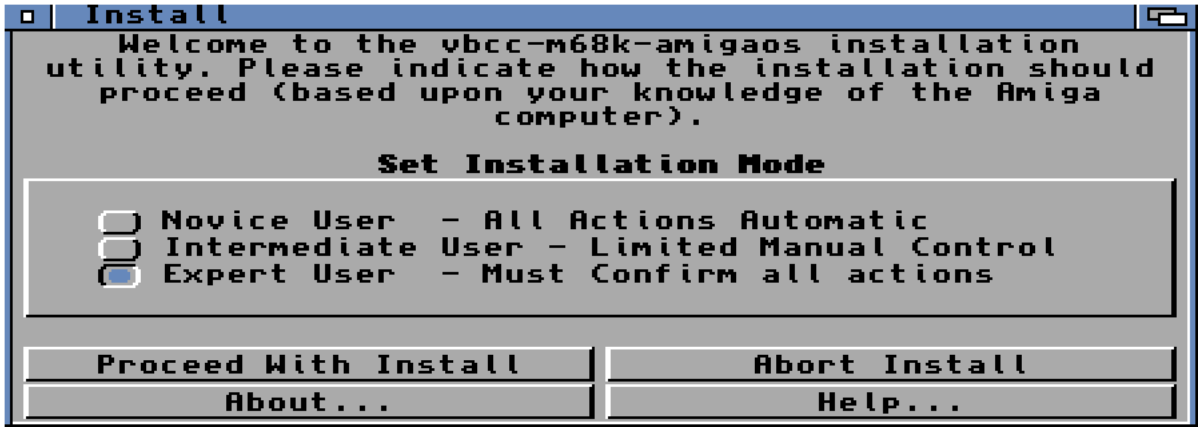
    - Specify the installation path as “**Development:**” when asked.



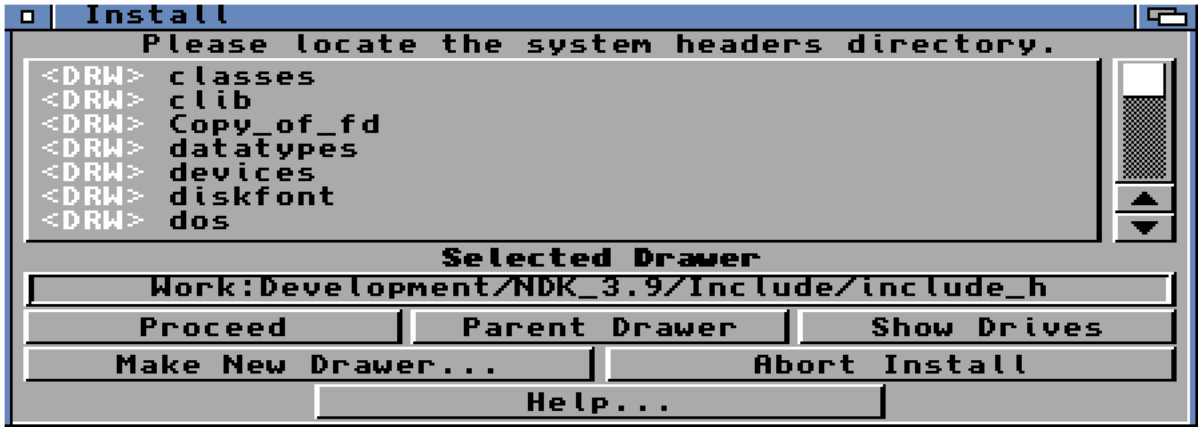
    - Proceed with the installation, allowing the installer to make necessary changes to the “user-startup”.



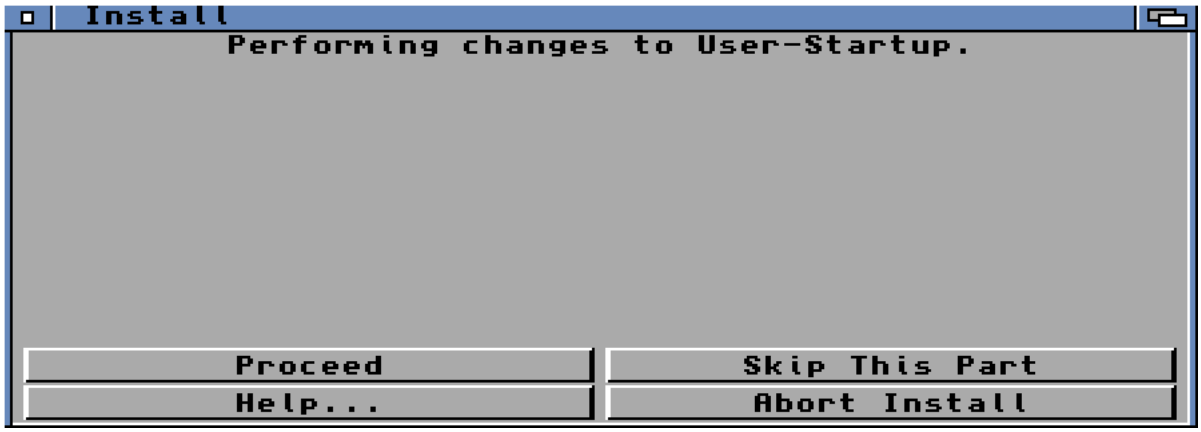
11. **Install VBCC Target:**
    - Go to “**Work:VbccSetupFiles/vbcc_target_m68k-amigaos/install**”.
    - Again, select "**Expert User**" mode.



    - When prompted for the path to includes, enter “**Development:NDK_3.9/include/include_h**”.



    - Proceed with the installation, confirming any changes to the “user-startup”.

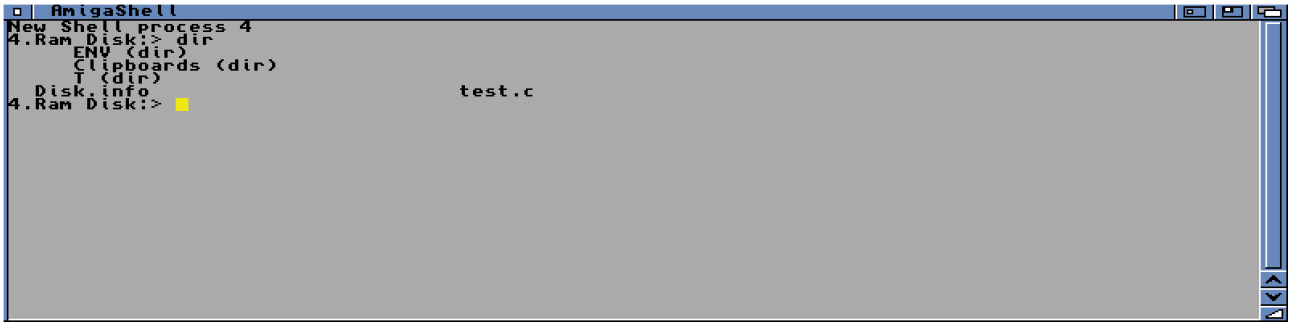


By completing these steps, you've successfully installed VBCC and its necessary components for Amiga development on your system. This setup positions you to begin coding and compiling projects for the Amiga platform.

Testing the Compiler with a Simple Program

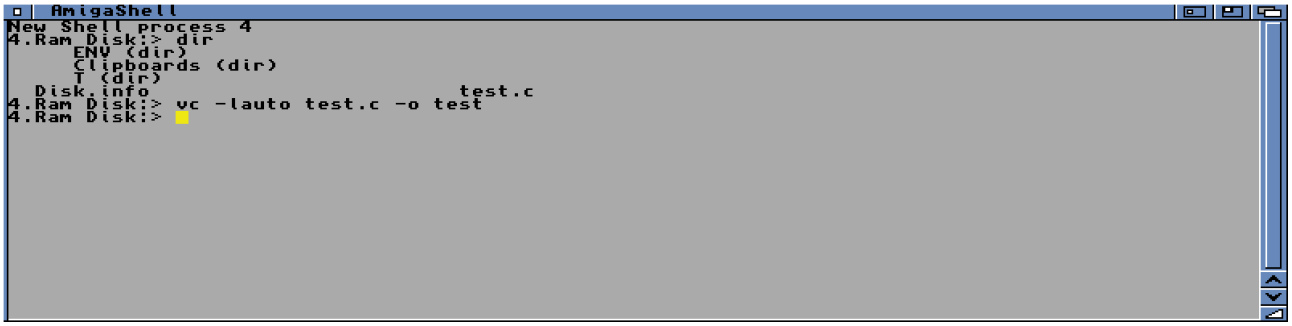
Now that you have installed and set up VBCC, it's time to test the compiler to ensure everything is functioning as it should. We'll do this by compiling a simple C program.

1. **Prepare the Test Program:** Copy the file "**test.c**" to the "**RAM:**" drive. The code listing for "**test.c**" can be found on page 13 of this document.



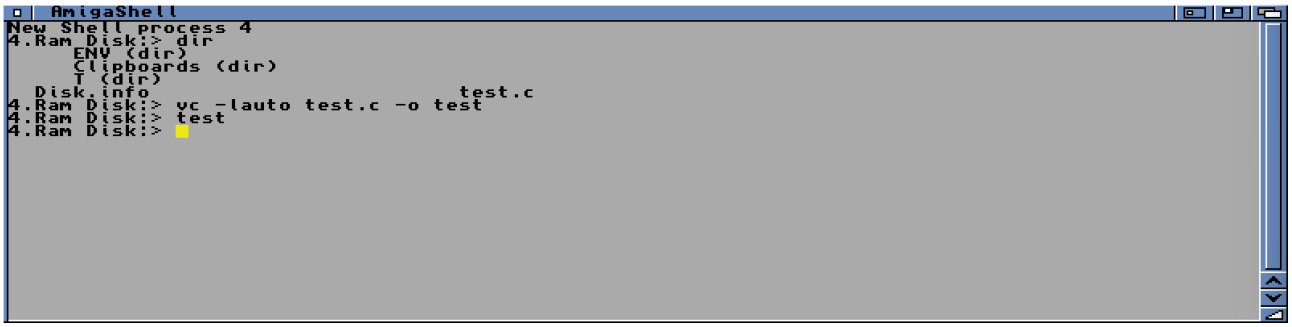
2. **Compile the Program:** Open a Shell window and navigate to the "**RAM:**" drive where you've placed "**test.c**". Type the following command to compile the program:

   vc -lauto test.c -o test

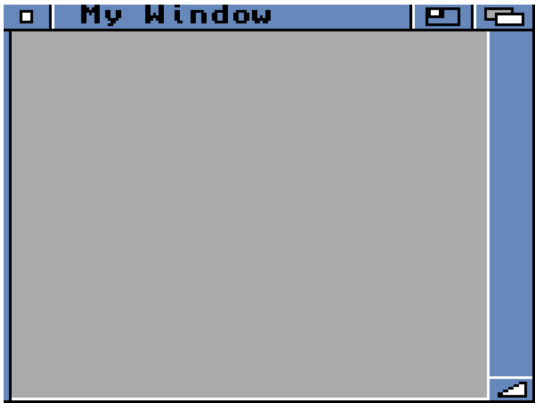


   This command tells VBCC to compile "**test.c**" into an executable named "**test**", linking libraries automatically with "**-lauto**".

3. **Verify Compilation:** After running the command, VBCC should compile the program without errors. To confirm everything went smoothly, you should see a new executable named "**test**" in the "**RAM:**" drive.



4. **Execution and Output:** Now, run the newly compiled program by typing “**test**” in the Shell window while still in the “**RAM:**” drive. You should expect to see a new window open. This window is the output of your program, indicating that the compilation was successful and the executable is running as expected.



This simple test confirms that VBCC is set up correctly on your system, and you're all set for diving into more advanced programming on your Amiga.

Test code listing

test.c

```
#include <proto/intuition.h>
#include <proto/dos.h>
#include <intuition/intuition.h>

int main() {
   struct Window *myWindow;

   struct NewWindow winlayout = {
      20, 20,
      200, 150,
      0,1,
      IDCMP_CLOSEWINDOW,
      WFLG_SIZEGADGET | WFLG_DRAGBAR | WFLG_DEPTHGADGET | WFLG_CLOSEGADGET | WFLG_ACTIVATE,
      NULL, NULL,
      "My Window",
      NULL,NULL,
      0,0,
      600,400,
      WBENCHSCREEN
   };

   myWindow = OpenWindow(&winlayout);
   Delay(200);
   if (myWindow) CloseWindow(myWindow);
   return(0);
   }
```