

ATOMIC CONSOLE DOCUMENTATION

INTRODUCTION

Atomic Console is a high-performance in-game command console for Unity. This documentation provides a comprehensive guide on how to use and integrate Atomic Console into your Unity projects.

TABLE OF CONTENTS

1. [Overview](#)
2. [Performance](#)
3. [Type Mapping](#)
4. [Command List](#)
5. [Multi-line Commands](#)
6. [Password Protection](#)
7. [Creating Commands](#)
8. [Custom Debugging](#)
9. [Console Skin](#)

OVERVIEW

Atomic Console is designed for performance and is capable of handling over 1000+ static and non-static commands while displaying up to 300 logs. It offers password-protected commands for added security and an IMGUI-based skin editor for customization. The console also supports auto-fill and command suggestion for quick command execution.

PERFORMANCE

To optimize performance, navigate to `Tools/AtomicConsole/Generate Assembly`. This tool scans all scripts to find commands and stores them in a list. If you add or delete commands, you must click `Generate Assembly` to update the list. Failing to do so will result in an error in the Unity console.

TYPE MAPPING

The Type Mapping tool is located under `Tools/AtomicConsole/TypeMapping`. This tool allows you to rename types like `float`, `bool`, etc., and even add your own type mappings.

COMMAND LIST

The Command List tool can be found under `Tools/AtomicConsole/CommandList`. This tool displays all available commands, their status (locked or static), and their descriptions. Each command has a toggle button that allows you to disable it without deleting it. After making changes, click `Apply`. You can also export the command list to a TXT file.

MULTI-LINE COMMANDS

Atomic Console supports multi-line commands. Commands can be separated by a semicolon (;). For example, `player.speed 5 ; player.jump 3`.

PASSWORD PROTECTION

To enable password protection for commands, navigate to the AtomicConsole GameObject. Enable the `Password Protected` toggle. Set a 4-digit password and click `Apply`. To access password-protected commands, type `>>` in the console, hit the space button, enter your password, and hit Enter.

CREATING COMMANDS

To create a command, add the `AtomicCommand` attribute to your function in your C# script. For example:

```
using AtomicConsole;

[AtomicCommand(name: "TestCommand")]
private void Test()
{
    Debug.Log("Hello Console");
}
```

After saving the script, click `Generate Assembly` to update the command list.

CUSTOM DEBUGGING

To use the custom debug tool, add the following line to your script:

```
using AtomicConsole.debug;
```

You can then debug using `AtomicDebug.Command("Hello Console");`.

CONSOLE SKIN

Atomic Console requires a Console Skin for its appearance. The asset comes with various skins, and you can select or create a new one.

ADDITIONAL FEATURES

- **Log Capacity:** Supports up to 300 logs, ideal for debugging and in-game data tracking.
- **User-Friendly Editor Window:** Allows you to disable commands instead of deleting them.
- **Low Learning Curve:** Requires only basic C# knowledge for setup and usage.
- **Sleep Mode:** Puts the console into a low-power state when not in use.
- **Lean Architecture:** Focuses only on essential features for an effective in-game console.
- **Auto-fill and Command Suggestion:** Offers real-time command suggestions. Hit the 'Tab' key to auto-fill the rest of the command.