Recast

Access Tools via MCP Server

Last Modified on 2026-11-18

MCP Server Overview

By adding a Right Click Tools MCP Server, you can run RCT actions as requests to your AI tools.

You can set up a Right Click Tools integration within GitHub Copilot Desktop, GitHub Copilot CLI and Visual Studio Code.

- GitHub Copilot Desktop Setup
- GitHub Copilot CLI Setup
- VS Code Setup

The Right Click Tools MCP Server is available to both RCT Enterprise customers and Community users. Enterprise customers will typically be able to run 75+ Right Click Tools actions, while Community users can access a subset of about 30 actions. See the full list of available actions.

Minimum Right Click Tools Version: 5.12.2608

NOTE: Right Click Tools Enterprise licenses are not subject to rate limiting. Community licenses are limited to 10 requests every 15 minutes.

For a walkthrough of MCP Server setup and use, see Jeff Scripter's video on our YouTube channel:Introducing the Recast Right Click Tools MCP Server.

Set Up MCP Server for GitHub Copilot Desktop

Prerequisites

- Right Click Tools Enterprise or Community v5.12.2608 or later installed
- GitHub account with at least the free tier of GitHub Copilot
- GitHub Copilot desktop app downloaded

Authorize the Right Click Tools MCP Server

To add the Right Click Tools MCP Server in GitHub Copilot:

1. Sign into GitHub.

2. In the **Device Activation** window, click **Continue** to sign in.

3. In the **Authorize your device** window, enter the code displayed in the app or on the device you're signing into.

4. Click **Authorize github**.

Add the Right Click Tools MCP Server

To add the Right Click Tools MCP Server in GitHub:

1. At the GitHub chat interface, open the**Customize** page.

2. On the **Installed** tab, click **Add server** and select **Add custom server** from the drop-down menu.

3. In the **Add MCP Server** window that opens, enter a **Server name**.

4. As the **Server type**, select **Local**.

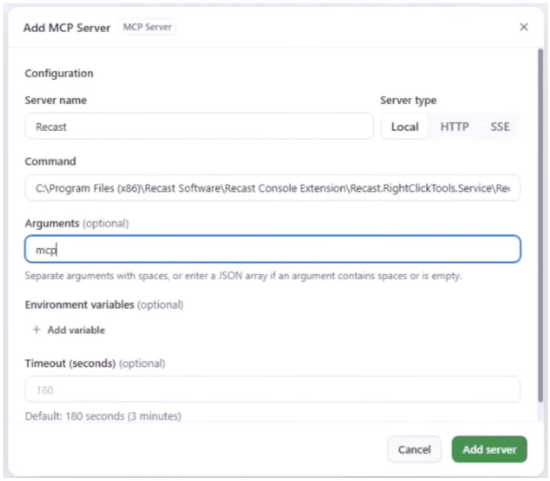
5. Navigate to **Program Files (x86)** > **Recast Software** > **Recast Console Extension**.

6. Right-click on the **Recast.RightClickTools.Service** and select **copy as path**.

7. Back in the **Add MCP Server** window, paste the path as the **Command**, removing the surrounding quotation marks.

8. Under **Arguments**, enter **mcp**.

9. Click **Add server**.



Set Up MCP Server for VS Code

Prerequisites

- Right Click Tools Enterprise or Community v5.12.2608 or later installed
- GitHub account with at least the free tier of GitHub Copilot
- GitHub Copilot desktop app downloaded

Add the Right Click Tools MCP Server

To add the Right Click Tools MCP Server in VS Code:

1. Open VS Code and sign into GitHub Copilot.

2. Press `F1` to open the Command Palette.

3. In the search bar, enter something like 'add mcp server' and select the**MCP: Add Server** command.



4. As the type of MCP Server to add, choose**Command (stdio)**.

5. Enter the path to Recast.RightClickTools.Service.exe:

`"C:\Program Files (x86)\Recast Software\Recast Console Extension\Recast.RightClickTools.Service\Recast.RightClickTools.Service.exe" mcp`

6. Enter a descriptive name, such as **RecastRightClickToolsService**.

VS Code will open your user profile mcp.json with a configuration similar to the following:

```
{
"servers": {
"RecastRightClickToolsMcp": {
"type": "stdio",
"command": "C:\\Program Files (x86)\\Recast Software\\Recast Console Extension\\Recast.RightClickTools.Service\\Recast.RightClickTools.Service.exe",
"args": [
"mcp"
]
}
},
"inputs": []
}
```

7. Start the MCP server by clicking **Start** above the server details.



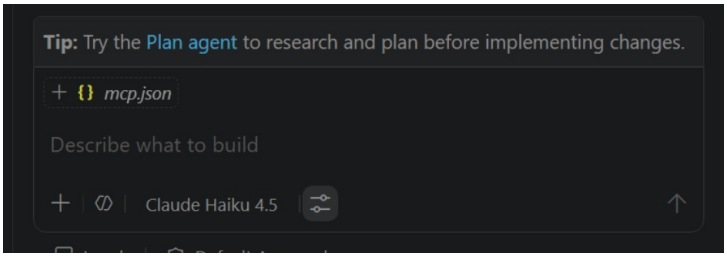
Startup can take up to 30 seconds while the tools load. If needed, you can select **More** > **Show Output** to review the log.

> The number of available tools depends on your Right Click Tools permissions, licensing, and configuration. Most RCT Enterprise users with full permissions will see 75+ tools, while RCT Community users typically see 30+.

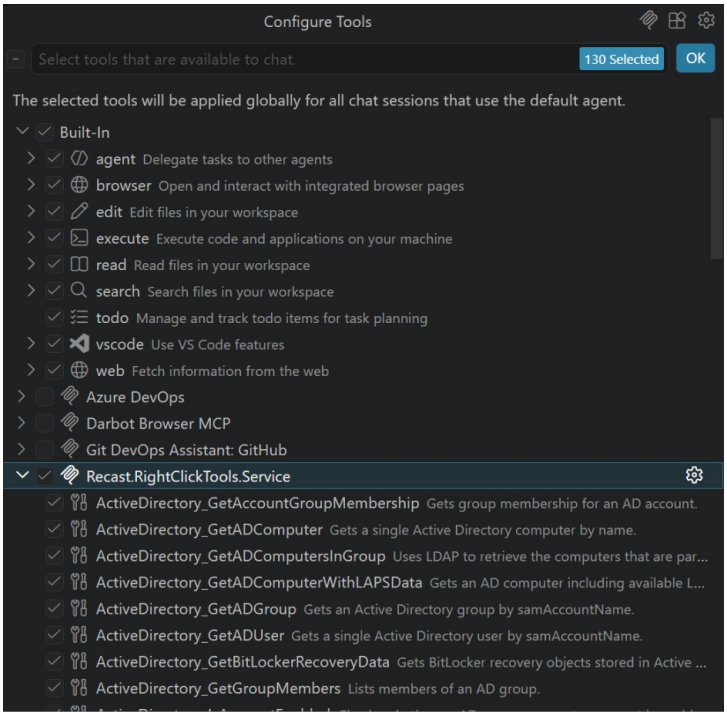
View/Manage Right Click Tools via the MCP Server

To view or manage tools:

1. Open the chat panel and select the Configure tools button (the sliders icon).



The **Configure Tools** panel that opens displays all MCP servers available in the session, along with which ones are enabled and which tools they expose.



When running MCP actions over a Recast Agent route, the device name in your MCP server request must match the device name that displays on the Recast Agents page in your Recast Management Server. Matching is case-sensitive. We strongly recommend entering all device names in upper case lettering, or creating a skill or instructions file for upper case device names.

Check That Your MCP Server is Enabled

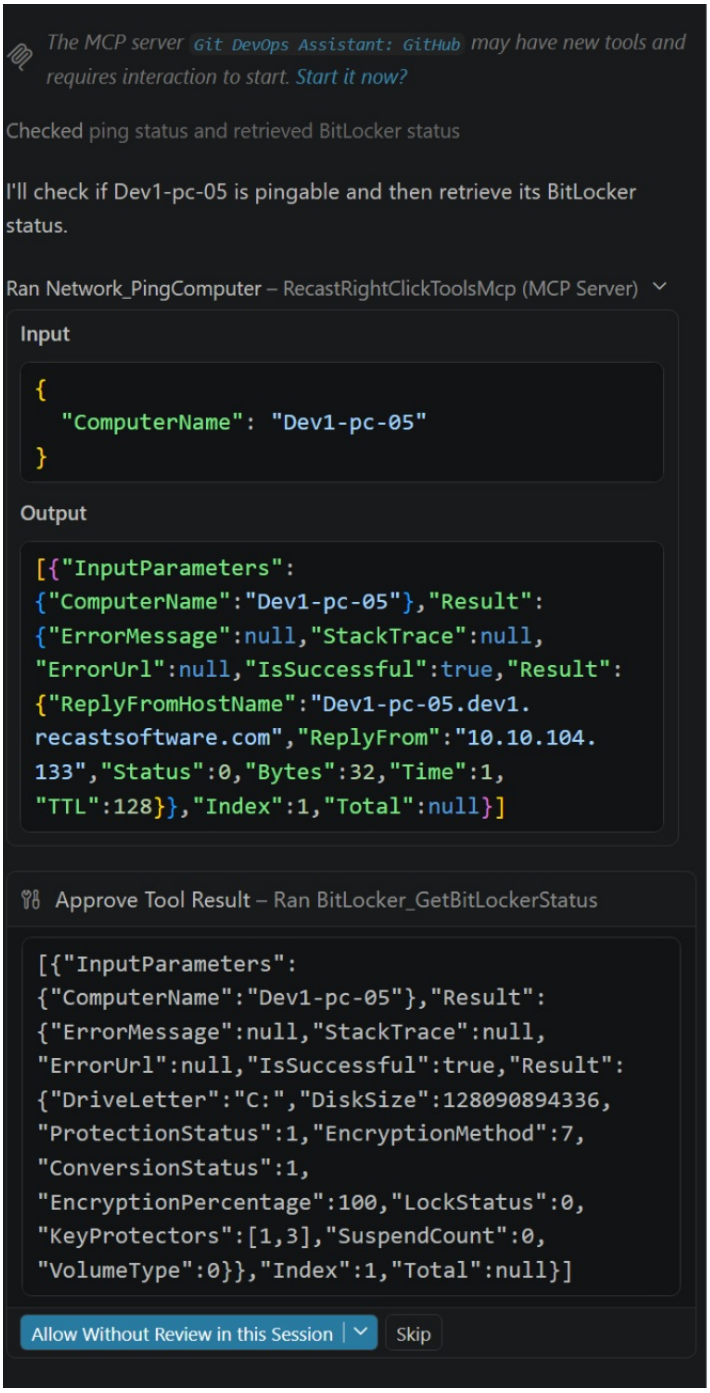
To verify that your RCT MCP server is enabled:

1. Click the gear icon to the right of the MCP server name to open mcp.json and confirm the configuration.

2. Run the sample prompt:

   'Can you check if Dev1-pc-05 is pingable and then get the list of bitlocker status of the device?'

You should see the tool name appear in light gray while Copilot is working. If the inputs look safe, approve the tool run.

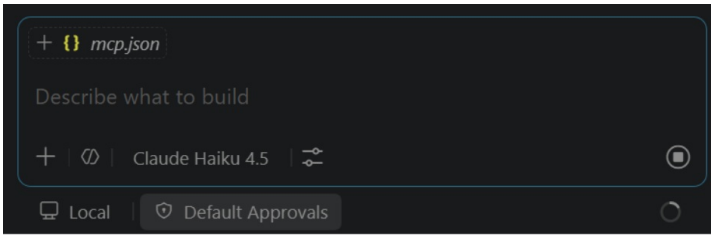


Enable Autopilot Approvals (Optional)

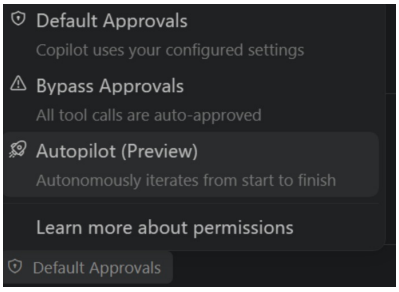
If you want fewer approval prompts, you can enable Autopilot approvals, but do so with caution.

To enable Autopilot approvals:

1. In mcp.json, click **Default Approvals**.



2. Select **Autopilot (Preview)**.



Set Up MCP Server for GitHub Copilot CLI

Prerequisites

- Right Click Tools Enterprise or Community v5.12.2608 or later installed
- GitHub account with at least the free tier of GitHub Copilot
- GitHub Copilot CLI installed

Add the Right Click Tools MCP Server

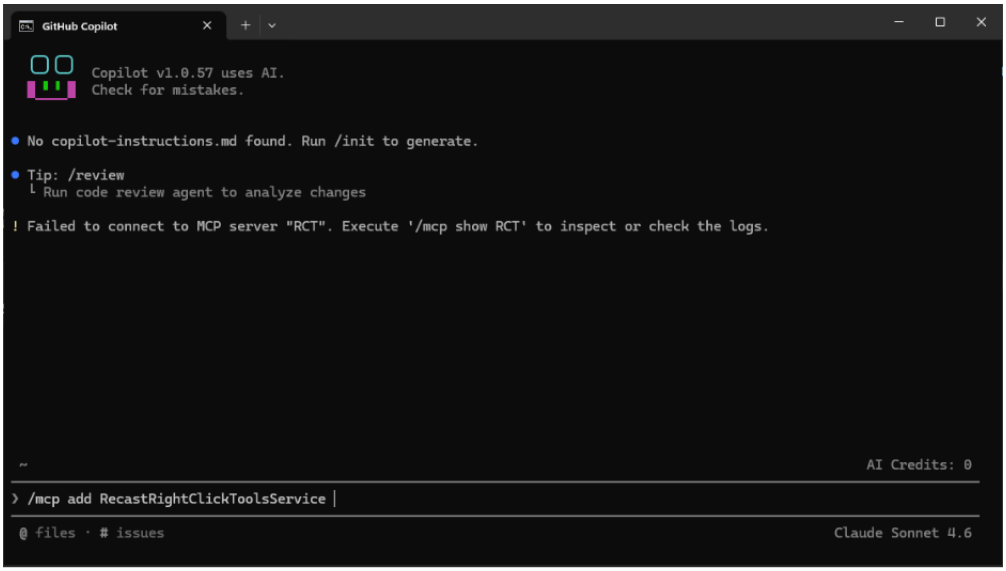
Before attempting to add the MCP Server, make sure Copilot CLI is installed and that you are signed in with your GitHub account.

To add the Right Click Tools MCP Server in Copilot CLI:

1. Open PowerShell or Command Prompt and run copilot to start Copilot CLI.

2. Sign in, if prompted.

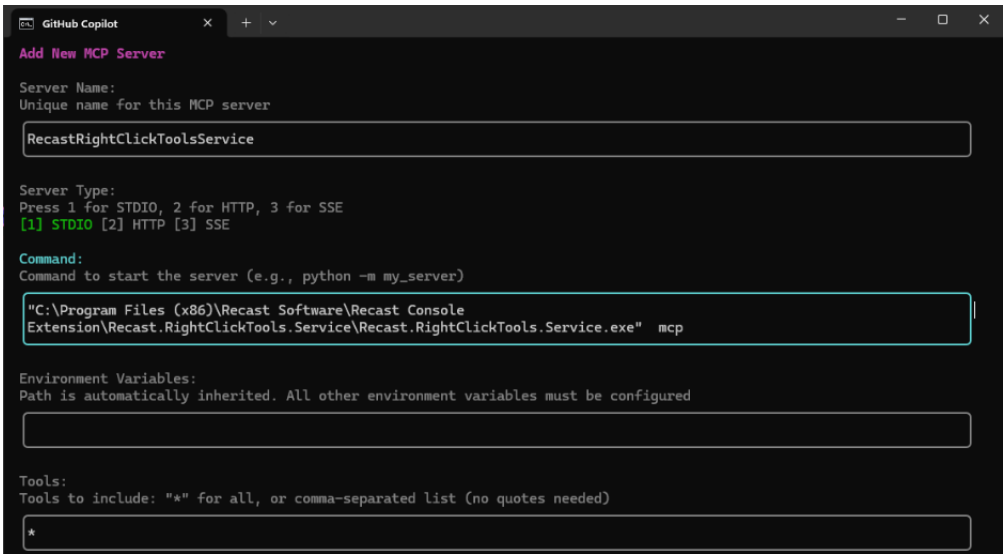
3. Run `/mcp add RecastRightClickToolsService`.



4. As the **Server Type**, choose 'STDIO'.

5. Enter the server command:

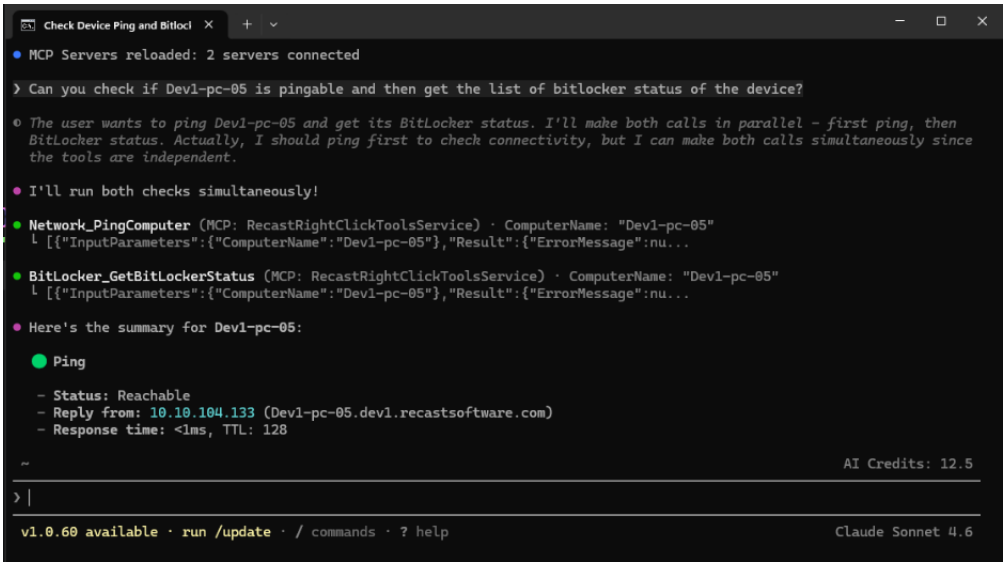
`"C:\Program Files (x86)\Recast Software\Recast Console Extension\Recast.RightClickTools.Service\Recast.RightClickTools.Service.exe" mcp`



Check That Your MCP Server is Enabled

To verify that your RCT MCP server is enabled, try running a sample prompt:

'Can you check if Dev1-pc-05 is pingable and then get the list of bitlocker status of the device?'



> *When running MCP actions over a Recast Agent route, the device name in your MCP server request must match the device name that displays on the Recast Agents page in your Recast Management Server. Matching is case-sensitive. We strongly recommend entering all device names in upper case lettering, or creating a skill or instructions file for upper case device names.*

Right Click Tools Available via MCP Server

The number of Right Click Tools actions available to you through the MCP server will depend on your Right Click Tools licensing, permissions, and configuration.

Most Right Click Tools Enterprise customers with full permissions will see all plugin actions, while RCT Community users typically have access to around 30 tools.

Full list of actions available via the RCT MCP server:

Active Directory

- `GetADComputer`
- `GetADComputerWithLAPSData`
- `GetADGroup`
- `GetADUser`
- `GetAccountGroupMembership`
- `GetBitLockerRecoveryData`
- `GetGroupMembers`
- `IsAccountEnabled`
- `SearchADComputers`
- `SearchADGroups`
- `SearchADUsers`
- `UnlockAccount`

Administration

Recast

- GetServerVersion
- GetExecutionHistory
- ListProxies
- ListAgents
- ListAgentGatewaysWithAgents
- ListMemcmServiceConnections
- ListActiveDirectoryServiceConnections
- ListAzureActiveDirectoryServiceConnections
- ListMbamServiceConnections

BitLocker

- GetBitLockerStatus
- GetRecoveryPasswordFromDevice

ConfigMgr Client

- AdvancedClientInformation
- Boundaries
- BoundaryGroups
- ClientCacheGeneralInformation
- ClientCacheInformation
- ClientExecutionHistory
- ClientInstalledComponents
- DeployedApplications
- Deployments
- GeneralClientInformation
- GetBoundaryInformation
- GetClientInformationCacheData
- GetConfigurationBaselines
- GetDeployedPrograms
- GetDeployedTaskSequences
- GetDeviceId
- GetServiceWindows
- GetUserPolicyEndpoint
- MissingSoftwareUpdates
- PackageInformation
- RunClientAction
- TaskSequenceInformation

ConfigMgr Server

- ContentDetails
- GetAllCollections
- GetAllDeviceCollections
- GetAllUserCollections
- GetBoundaries
- GetBoundaryGroups
- GetCollectionVariablesforDevice
- GetCollectionsforDevice
- GetCollectionsforUser
- GetContentStatus
- GetDeviceCollectionFolders
- GetDeviceCollectionInformationforDevice
- GetDeviceCollectionMembers
- GetDistributionPointGroupStatusforPackage
- GetDistributionPointStatusforPackage
- GetWarrantyInformation
- GetComputerWarranty

- `GetComputerWarrantyByComputerName`
- `GetRecentDevicesScanned`
- `GetDeviceCount`
- `GetManufacturerCounts`
- `GetDevicesWithWarrantyExpiringSoon`

Event Log

- `GetEventLogEntries`
- `GetEventLogInfo`
- `GetEventLogs`

Filesystem

- `FindLogFiles`
- `GetLogEntries`

MBAM

- `GetMBAMCompliance`
- `GetMBAMComplianceForAllMachines`
- `GetMachinesWithMBAMKeys`
- `GetMachinesWithMBAMKeys_v2`
- `GetRecoveryKeysForDevice`
- `GetTPMHash`
- `GetTPMHashForUser`

Networking

- `PingComputer`

Registry

- `GetValue`
- `ListValues`
- `ListSubkeys`
- `ListHives`

System Information

- `GetAllLocalGroupMembers`
- `GetBatteryInformation`
- `GetDiskSpaceInfo`
- `GetDrivers`
- `GetGeneralInformation`
- `GetLoggedInUsers`
- `GetRunningProcesses`
- `GetUpdates`
- `GetUserProfileSize`
- `GetUserProfiles`
- `GetUserSessions`
- `SystemInformationCollection`

Windows Security

- `GetTpmStatus`
- `GetWindowsFirewallProfiles`
- `GetWindowsFirewallRules`