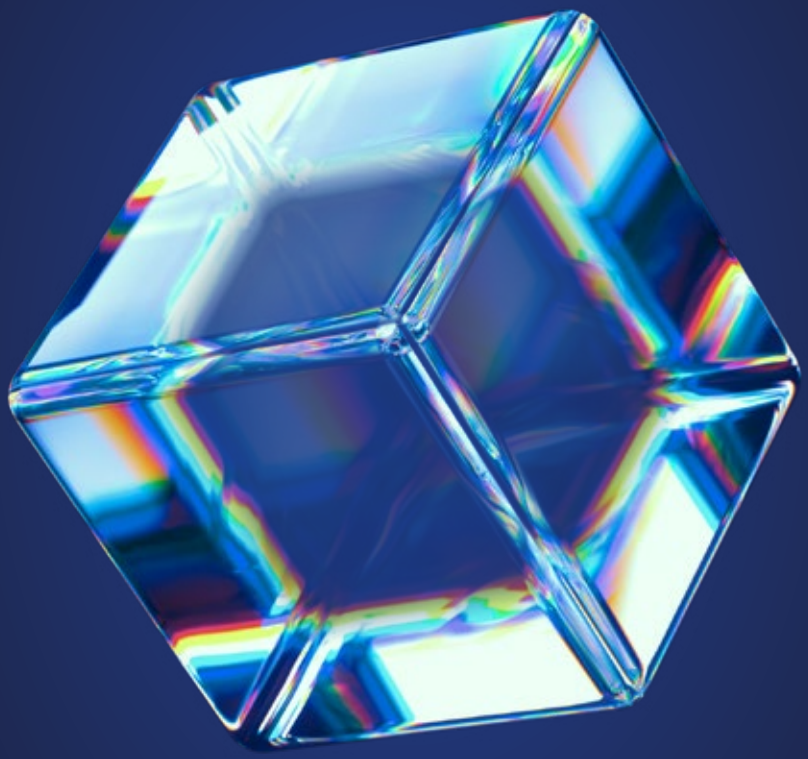
BlockDAG

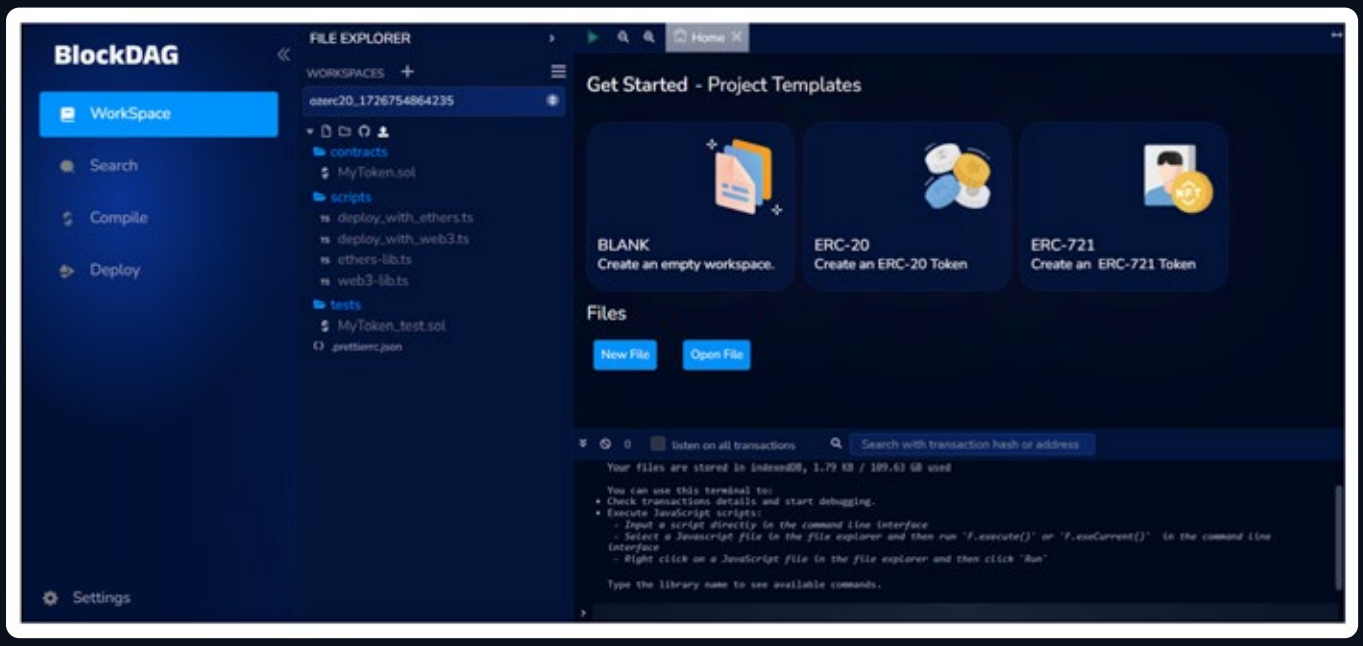


User Guide for **BlockDAG IDE**

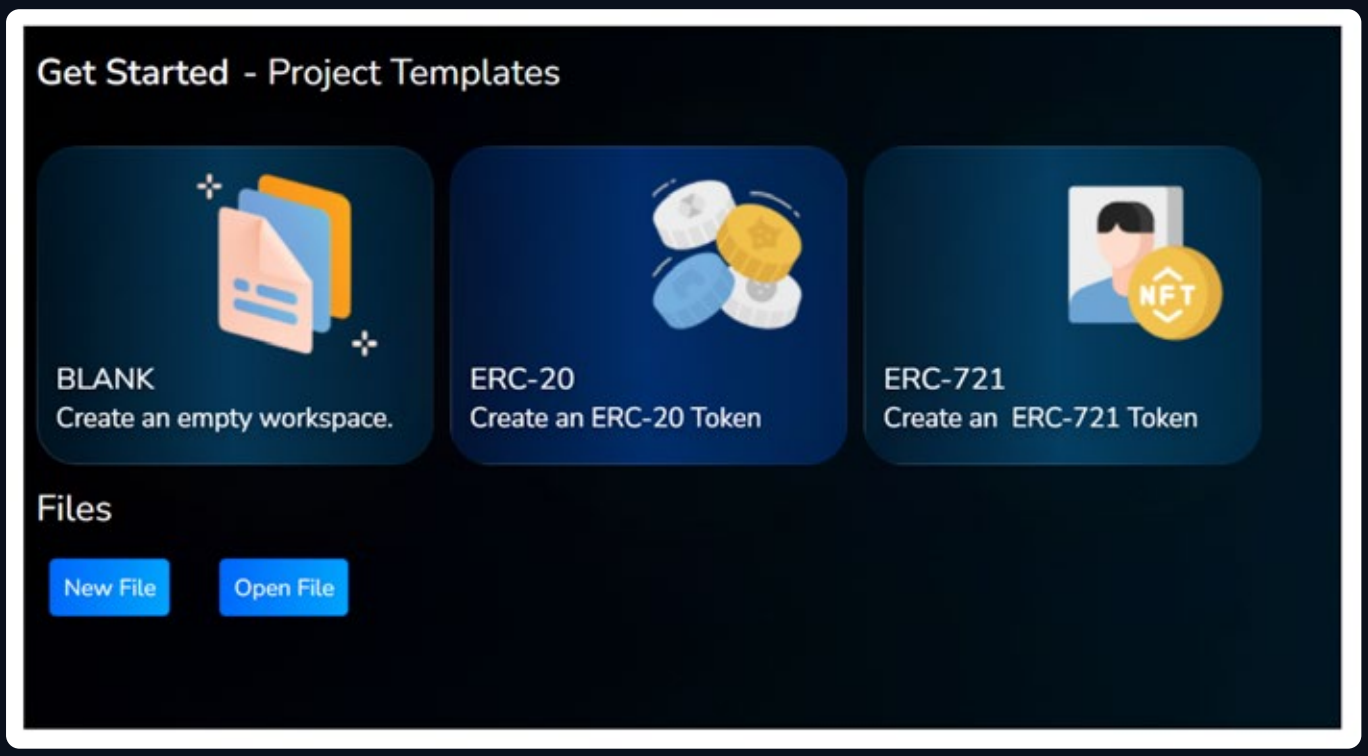
Date: 20-09-2024

BlockDAG

Ide Layout



Icon panel: it allows the users to switch between different plugins that are displayed. By clicking on the panels users can change the plugin to the one they want to use.



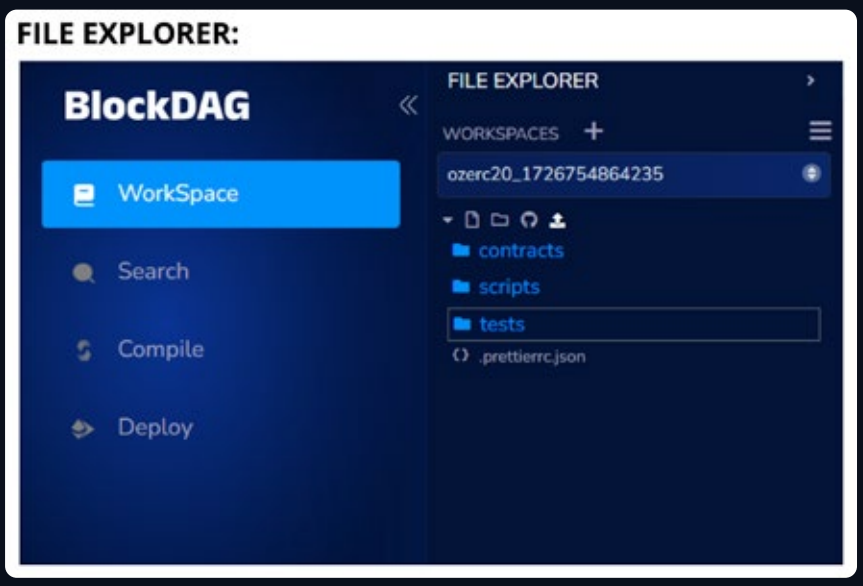
Side dashboard: Most plugins will have their GUI displayed here.
Main panel: Earlier the main panel was used to edit the plugins. However, the new version of IDE provides the feature to accommodate both plugins and files for IDE to compile.
Terminal: Like all the other CLI, this terminal also provides the feature to view the outcomes of their interactions with the GUI. It also allows the users to run scripts.

BlockDAG

Home Tab

This tab is accessible in the main panel.
By using the toggle button , a user can switch between different templates or open new files from the files section.

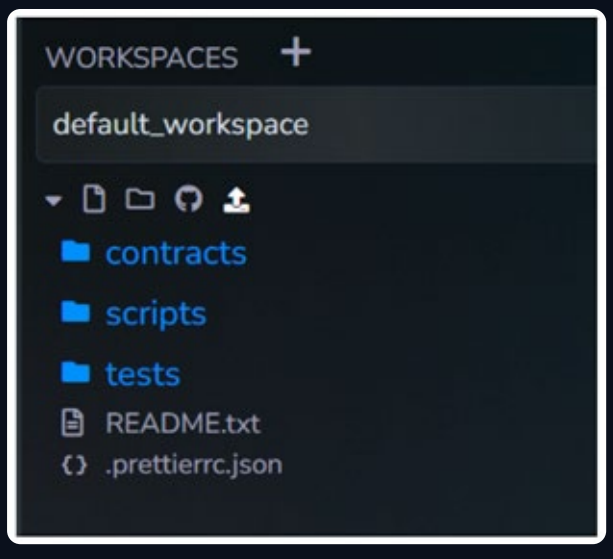
FILE EXPLORER:

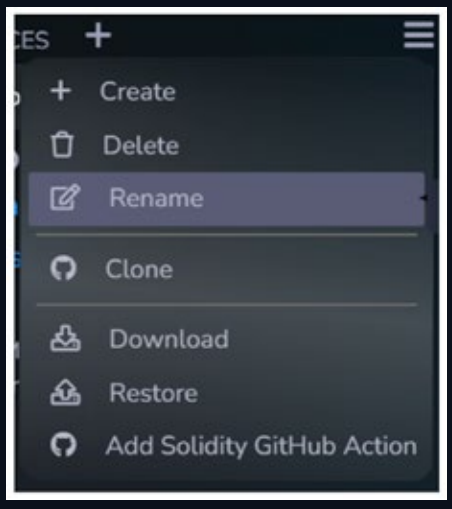


Click on the workspace option on the left panel to access the file explorer module. In the file explorer, users can manage their workspace and files, and can also access the context menu by right clicking on the folder. This menu provides options for users to perform various actions on their files and folders such as renaming, copying or deleting files.

FILE EXPLORER GUIDE:

Workspaces basically allows the user to better manage their files by enabling you to separate projects. You can perform the following functions through this.





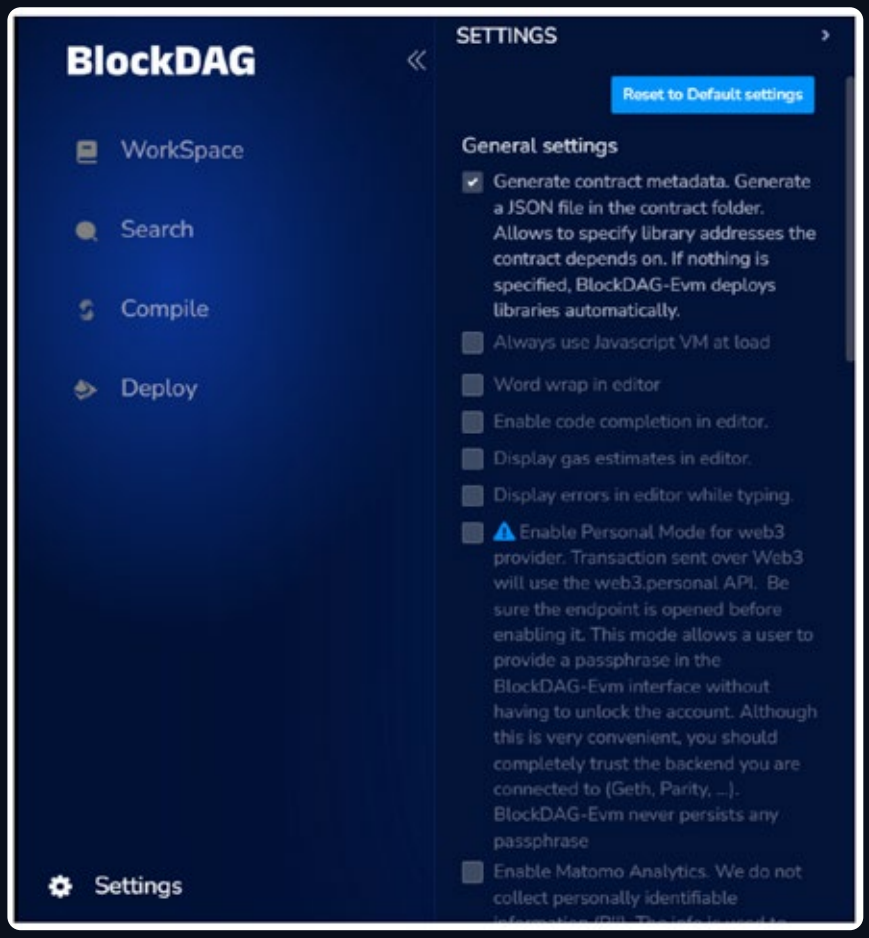
- Add a workspace
- Dropdown menu create, delete, rename or download a particular Workspace.
- Choose a Workspace.
- Expand button
- Create a file
- Create a folder
- Publish the Workspace to a GIST
- Load a file into the current Workspace
- By clicking on the drop-down menu,you can access the following options

BlockDAG

Home Tab

SETTINGS:

Workspaces basically allows the user to better manage their files by enabling you to separate projects. You can perform the following functions through this.



- To access the settings option, click on the “setting” on the bottom of the left panel.
- You can choose the option of reset to default setting to get back the settings as they were originally.
- **Word wrap** setting allows to wrap the text or not
- **Personal mode**, this mode can be used while connecting to a local node.

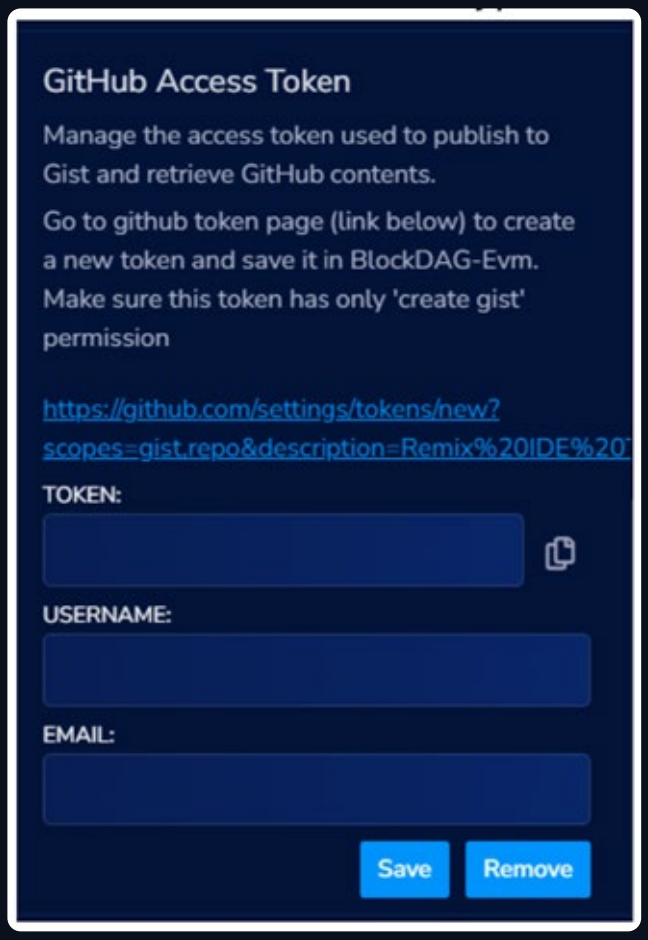
BlockDAG

Home Tab

GITHUB ACCESS TOKEN:

To carry out git operations on github and create GISTS, it is necessary to provide the access token. It provides specific permissions to execute git commands. Depending upon the operations entered, the user might need to enter the username and email address.

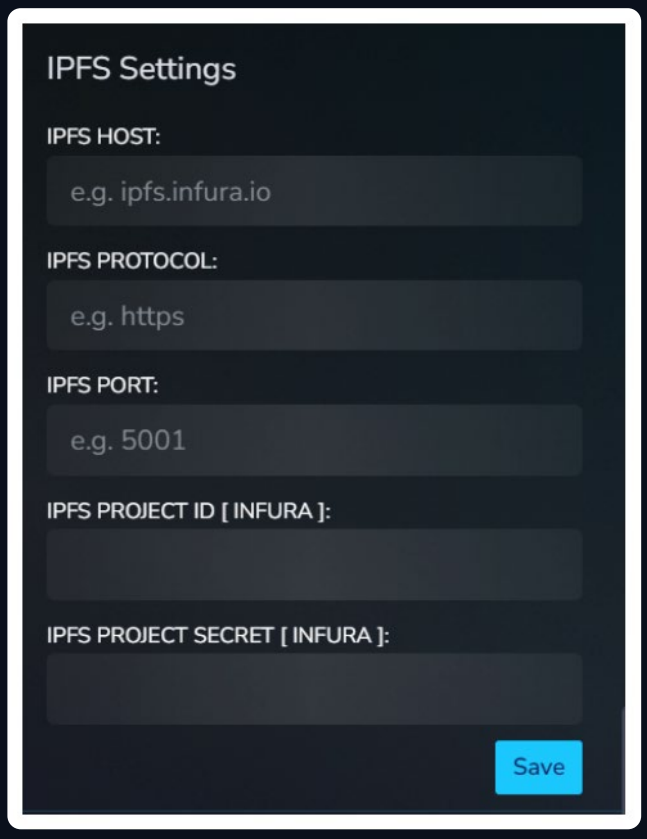
Please note: the IDE does not store any password information.



BlockDAG

Home Tab

IPFS SETTING:



These above details are necessary to deploy the contract. If the details are not provided, then you'll be using the INFURA mode which does not persist the data.

Steps to Deploy a Smart Contract through BlockDAG IDE:

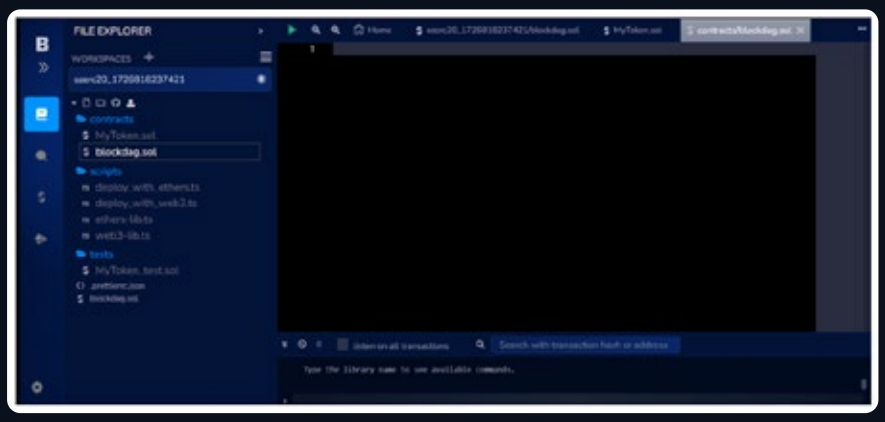
Step 1:
Access the BlockDAG IDE Go to https://ide.blockdag.network/

Step 2:
Create a New File in the Workspace

- From the left-panel menu, navigate to **“Workspace”** and click on the **“Create New File”** icon.
- Name the new file **“blockdag.sol”** under the **“contracts”** folder.
- A new file will be generated, and the main editor panel will open with ablank file.

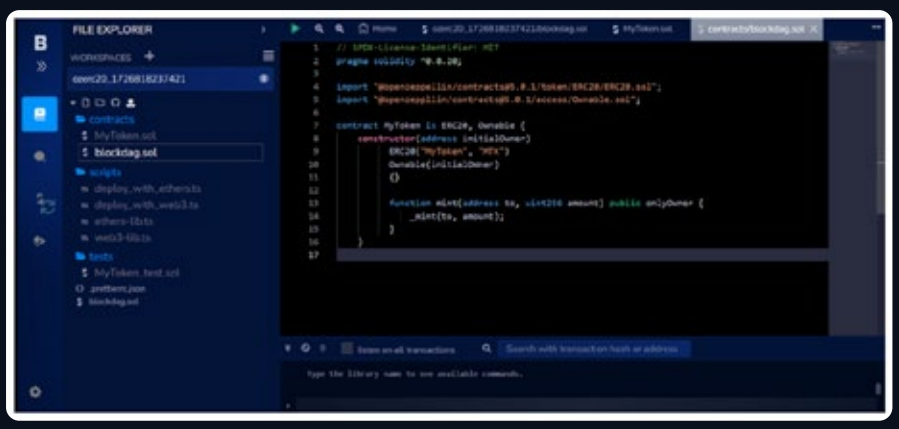
Step 3:
Write or Import a Solidity Contract

- Write or import your Solidity contract into the **“BlockDAG IDE”**.



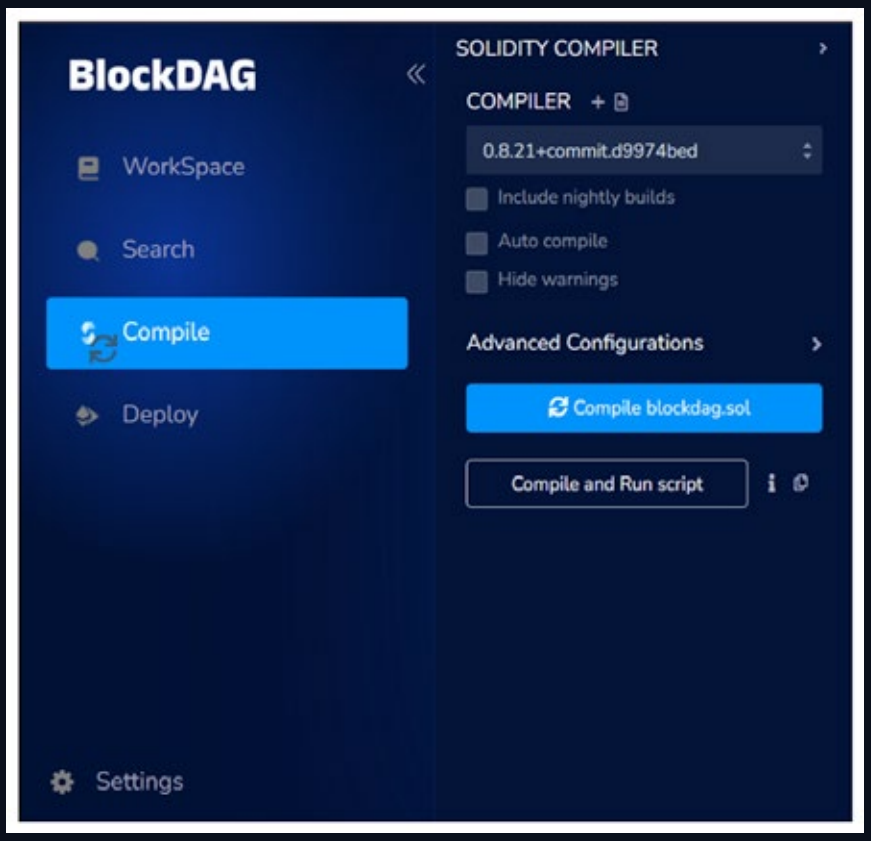
Step 4:
Compile the Smart Contract

- Configure the Solidity compiler settings within the **“BlockDAG IDE”**.
- Select your Solidity contract file (blockdag.sol) and click on **“Compile”**.
- Upon successful compilation, your contract will be ready for deployment. You should see a confirmation of a successful compile on your screen.



Step 5:

Compilation – Configure the Solidity compiler settings within BlockDAG IDE



- Click on **'Compile'** for the **'blockdag.sol'** file to compile the contract.
- Upon successful compilation, your screen will resemble the following:

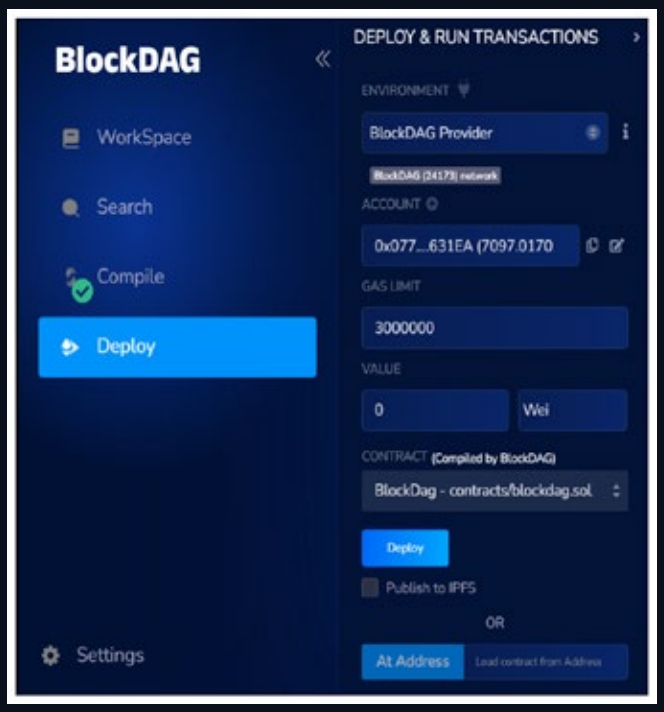
Step 6:

Deployment

Navigate to the **'Deploy & Run Transactions'** sidebar in the left panel.

To utilise this module, a compiled contract is necessary. If there is a contract name in the 'CONTRACT' select box (located under the 'VALUE' input field), you can proceed with using this module.

- **BlockDAG Provider:** For connecting BlockDAG IDE to an injected web3 provider. The most common injected provider is MetaMask
- **Hardhat Provider:** For connecting BlockDAG IDE to a local Hardhat test chain.
- **Ganache Provider:** For connecting BlockDAG IDE to a local Truffle Ganache test chain.



- **Click 'Deploy' and confirm the transaction on the metamask.**