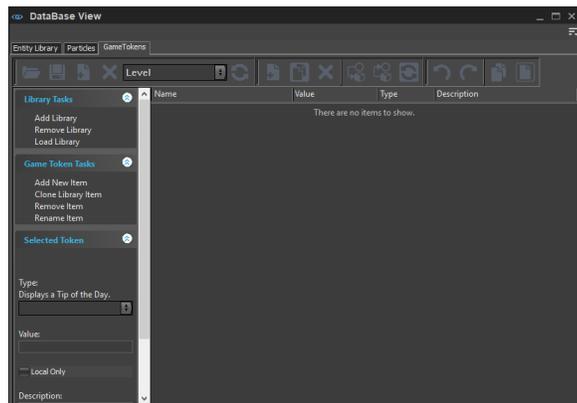


GameTokens tab on DataBase View is no longer supported in CRYENGINE. It will be removed from the engine layout in the future and some features might not function as expected.

A Game Token is a script object or variable that is used for storing values. These tokens can be used for performing simple logic manipulations, and checking a value in the game scripting language.

- [Toolbar](#)
- [Creating Tokens](#)
- [Saving Tokens across Levels](#)
- [Debugging](#)



Possible token functions include states (squad mate = dead or alive), characters, information (research tool results), sequential logic (an event has taken place, in if/then sequences), or objects (for example, weapons, vehicles).

These are represented within various flow nodes with different functions, inputs, outputs, and checks.

In the database, all the defined values are stored in libraries. By loading or not loading the same library for every level, tokens can be set to be active only within one level, across multiple levels, or only in part of one level.

## Toolbar

Game Tokens tab uses the default DataBase View toolbar set. For all the toolbar options and their functionalities, please refer to the [DataBase View](#) page.

## Creating Tokens

In order to create a new game token, click **Add New Item** on Game Token Tasks list and give the token a name. After you have created a token, you can choose from the following types: **Bool**, **Int**, **Float**, **String**, or **Vec3**.

Depending on the chosen type, the token will get the following values:

Value	Description
<b>Bool</b>	True or False states.
<b>Int</b>	Any whole number.
<b>Float</b>	Any floating point number.
<b>String</b>	Any sequence of characters.
<b>Vec3</b>	Any 3 dimensional positional coordinate (x,y,y) - for example, 290,200,180.

Don't forget to add a description in the Description field, you may find it come in handy later on when you have many tokens in the library.

## Saving Tokens across Levels

There are 2 specific Flow Graph nodes related to the storing of information across levels.

- **Mission:GameTokensLevelToLevelStore** - Trigger this at the end of the level, and it will save the values specified in the listed game tokens.
- **Mission:GameTokensLevelToLevelRestore** - Trigger this at the beginning of the next level to retrieve the values from the previous level.

## Debugging

