Context

Last Updated: 2021-03-05

Context is a data type that you can use to store sets of data.

Contexts are like the `struct` data type in C/C++. Contexts can be used to store elements of any combinations of data types, including other contexts and arrays. This data is stored in a set of variables called member variables that are "contained" inside the context. Member variables can be of any type, including other contexts.

You reference member variables using the dot notation. This is also the way that you reference member variables in a `struct` in languages like C and C++. In this notation, you specify the name of the context and the name of the member variable separated by a period (.). You use this notation when you assign values to member variables and when you reference the variables elsewhere in a policy.

> **Important:** A built-in context is provided, called the policy context, that is created automatically whenever the policy is run. The policy context contains all of the variables used in the policy, including built-in variables.

Unlike arrays and scalar variables, you must explicitly create a context using the `NewObject` function before you can use it in a policy. You do not need to create the member variables in the context. Member variables are created automatically the first time you assign their value.

The following example shows how to create a new context, and how to assign and reference its member variables:

```
MyContext = NewObject();
MyContext.A = "Hello, World!";
MyContext.B = 12345;

Log(MyContext.A + ", " + MyContext.B);
```

This example prints the following message to the policy log:

```
Hello, World!, 12345
```

The following policy shows how to create a context called `MyContext` and assign a set of values to its member variables.

```
MyContext
= NewObject();

MyContext.One = "One";
MyContext.Two = 2;
MyContext.Three = 3.0;

String1 = MyContext.One + ", " + MyContext.Two + ", " + MyContext.Three;

Log(String10;
```

When you run this policy, it prints the following message to the policy log:

```
One, 2, 3.0
```