

Logging and Error Functions

Logging and error functions can and should be used to provide appropriate reactions to script actions.

GetLastError

Declaration

```
string GetLastError()
```

Syntax

```
GetLastError()
```

Description

Returns the last error reported in a script or transformation. It can be used with the [ResetError\(\)](#) and [Eval\(\)](#) functions for error handling.

Note that the `GetLastError()` function does not carry the error message over into another operation. To do that, Jitterbit variables are available that contain that information for use between operations:

- `$jitterbit.operation.error`
- `$jitterbit.operation.previous.error`
- `$jitterbit.operation.last_error`

See also the [RaiseError\(\)](#) function.

Examples

```
// Discard previous errors
ResetLastError();
error = "";

// On failure, put the last error into a variable
Eval(orderType = DBLookup("<TAG>endpoint:database/My Database</TAG>",
  "SELECT ORDER_TYPE FROM PO_HEADER WHERE PO_NUMBER = " + PO_NUM),
  error = GetLastError());

// If an error, raise it with the error in the message
If(Length(error) > 0,
  RaiseError("Failed to lookup Order Type: "
    + error), orderType);
```

[\[Back to Top\]](#)

RaiseError

Declaration

```
void RaiseError(string message)
```

Syntax

```
RaiseError(<message>)
```

Required Parameters

On This Page

- [GetLastError](#)
- [RaiseError](#)
- [ResetLastError](#)
- [SetLastError](#)
- [WriteToOperationLog](#)

Search in This Topic

Related Articles

- [Activities](#)
- [Jitterbit Log File Locations](#)
- [Editing the Configuration File - jitterbit.conf](#)
- [Operation Options \(Cloud Studio\)](#)
- [Operation Options \(Design Studio\)](#)

Related Topics

- [Cloud Studio](#)
- [Functions](#)
- [Scripts](#)

Last updated: Oct 04, 2019

- **message**: A string message

Description

Causes a script or transformation to fail, and the contents of the parameter `message` to be displayed in the error log.

The entire text written to the error log will be:

```
Fatal Error
<message>
This error was raised by a call to the RaiseError function.
```



WARNING: The `RaiseError()` function should be used as a last-resort, as it raises a fatal error. If a script raises it, there is no control by the calling program on what is produced to the client. Jitterbit does not have a "raise-and-capture" exception mechanism, and raising this error will result in a Server Error with a Status of 500. Instead, programs should test for inputs and outputs and handle conditions defensively.

Examples

```
message = "The source field "
  + "'Price' has an invalid value. Exiting.";
RaiseError(message);
```

[\[Back to Top\]](#)

ResetLastError

Declaration

```
void ResetLastError()
```

Syntax

```
ResetLastError()
```

Description

Sets the last error to an empty string. This is identical to calling `SetLastError(" ")`. See also the function [SetLastError\(\)](#).

Examples

```
// Reset the last error
ResetLastError();
```

[\[Back to Top\]](#)

SetLastError

Declaration

```
void SetLastError(string message)
```

Syntax

```
SetLastError(<message>)
```

Required Parameters

- **errorMessage**: A string message

Description

Sets a user-defined last error. The message will be logged as a warning and the `GetLastError()` function will return the message unless another error occurs. See also the function `ResetLastError()`, which performs the same action of setting the last error but without logging a message.

Examples

```
// Sets the last error to a message  
message = "Invalid value: source field 'Price'";  
SetLastError(message);
```

[\[Back to Top\]](#)

WriteToOperationLog

Declaration

```
string WriteToOperationLog(string message)
```

Syntax

```
WriteToOperationLog(<message>)
```

Required Parameters

- **message**: A string message

Description

Writes a message to the operation log.

Examples

```
// Write a message to the Operation Log  
WriteToOperationLog("The source field 'Price' has an invalid value.");
```

[\[Back to Top\]](#)