EIP 211: New opcodes: RETURNDATASIZE and RETURNDATACOPY

Return to Ethereum ERCs

> **Note**: The following is an excerpt from the official Ethereum site. It is provided here as a convenience and is not authoritative. Refer to the original document as the authoritative reference.

Table 1: Data sheet for New opcodes: RETURNDATASIZE and RETURNDATACOPY

Title	New opcodes: `RETURNDATASIZE` and `RETURNDATACOPY`
Author	Christian Reitwiessner
Status	Final
Created	2017-02-13
Description	http://eips.ethereum.org/EIPS/eip-211
Specification	http://eips.ethereum.org/EIPS/eip-211#Specification
Category	Core
Replaces	EIP 5

Simple Summary / Abstract

A mechanism to allow returning arbitrary-length data inside the EVM has been requested for quite a while now. Existing proposals always had very intricate problems associated with charging gas. This proposal solves the same problem while at the same time, it has a very simple gas charging mechanism and requires minimal changes to the call opcodes. Its workings are very similar to the way `calldata` is handled already; after a call, return data is kept inside a virtual buffer from which the caller can copy it (or parts thereof) into memory. At the next call, the buffer is overwritten. This mechanism is 100% backwards compatible.

Motivation

In some situations, it is vital for a function to be able to return data whose length cannot be anticipated before the call. In principle, this can be solved without alterations to the EVM, for example by splitting the call into two calls where the first is used to compute only the size. All of these mechanisms, though, are very expensive in at least some situations. A very useful example of such a worst-case situation is a generic forwarding contract; a contract that takes call data, potentially makes some checks and then forwards it as is to another contract. The return data should of course be transferred in a similar way to the original caller. Since the contract is generic and does not know about the contract it calls, there is no way to determine the size of the output without adapting the called contract accordingly or trying a logarithmic number of calls.

Compiler implementors are advised to reserve a zero-length area for return data if the size of the return data is unknown before the call and then use `RETURNDATACOPY` in conjunction with `RETURNDATASIZE` to actually retrieve the data.

***Note** this proposal also makes the EIP that proposes to allow to return data in case of an intentional state reversion (EIP-140) much more useful. Since the size of the failure data might be larger than the regular return data (or even unknown), it is possible to retrieve the failure data after the CALL opcode has signalled a failure, even if the regular output area is not large enough to hold the data.*