Ruby master - Feature #16129

Call initialize_clone with freeze: false if clone called with freeze: false

08/26/2019 04:23 AM - jeremyevans0 (Jeremy Evans)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		

Description

This makes it possible for initialize_clone to correctly not freeze internal state if the freeze: false keyword is passed to clone.

If clone is called with freeze: true or no keyword, do not pass a second argument to initialize_clone to keep backwards compatibility.

This makes it so that external libraries that override initialize_clone but do not support the freeze keyword will fail with ArgumentError if passing freeze: false to clone. I think that is better than the current behavior, which succeeds but results in an unfrozen object with frozen internals.

The attached patch implements this support and fixes related issues in set and delegate in stdlib.

Related issues:

Related to Ruby master - Bug #14266: Set#clone(freeze: false) makes frozen in...	**Closed**

History

#1 - 08/26/2019 04:23 AM - jeremyevans0 (Jeremy Evans)

- Related to Bug #14266: Set#clone(freeze: false) makes frozen internal hash added

#2 - 07/28/2020 08:11 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Closed

Files

initialize-clone-freeze-false.patch	8.1 KB	08/26/2019	jeremyevans0 (Jeremy Evans)