

# Off-Heap Annotations

## 1 Introduction

This document describes the custom off-heap annotation types to be used in GemFire source code. These annotations codify the standard nomenclature when documenting source that is affected by off-heap.

Each annotation will include the following information:

- Targets - what Java language elements are legal for the annotation
- Retention
  - Source - annotation is discarded by the compiler
  - Class - annotation is stored in the class file, but ignored by the vm
  - Runtime - annotation is stored in the class file and read by the vm
- Members - what information the annotation can store. All annotation members will be optional.
- Examples - annotation usage samples

### 1.1 @Documented

All annotations will be documented meaning that they will use `@Documented` in their definition (annotation.java file). This means that they will show up in GemFireXD developer (not end user) JavaDocs (in addition to the source code). Additionally, they can be subject to Doclet processing for any additional documentation artifacts (such as retain/release and identity reports).

## 2 OffHeapIdentifier

This Java enumeration is used to uniquely identify annotation groups within the GemFireXD source code. Annotations groups are off-heap annotations that relate to one another through the artifacts they are annotating. For example, `EntryEventImpl` might define one or more annotation identifiers for both the old and new region entry values. These identifiers are then used in the annotations that decorate these two entry values, which provide a logical grouping.

[blocked URL](#)

## 3 @Offheap

This annotation is used to mark fields, local variables, parameters and types as potential off-heap references that require `retain()` and `release()` to manage their lifecycle. When marking a type, such as `EntryEventImpl`, this annotation indicates that the type manages one or more off-heap references as part of its lifecycle.

### 3.1 Targets

- Field (instance variable)
- Local Variable
- Parameter
- Type (class or interface declaration)

### 3.2 Retention

- Source

### 3.3 Members

- value - Type is `OffHeapIdentifier[]`. Optional. Used to match this `@Offheap` with one or more additional off-heap annotations.

### 3.4 Examples

1. `@Offheap` on a field with multiple identifiers

[blocked URL](#)

2. `@Offheap` on a local variable with no identifiers

[blocked URL](#)

## 4 @Retained

This annotation is used for two purposes:

1. When used on a method declaration it indicates that the method called retain on the return value if it is an off-heap reference.
2. When used on a constructor declaration this annotation indicates that field members may be off-heap references and retain will be invoked on the field methods. This also indicates that the class will have a release method.
3. When used on a parameter declaration it indicates that the method will call retain on the parameter if it is an off-heap reference.

### 4.1 Targets

- Parameter
- Method
- Constructor

### 4.2 Retention

- Source

### 4.3 Members

- value - Type is OffHeapIdentifier[]. Optional. Used to match this @Retained with one or more off-heap annotations.

### 4.4 Examples

4.4.1 @Retained on a method parameter

[blocked URL](#)

4.4.2 @Retained on a method declaration

[blocked URL](#)

4.4.3 @Retained with an OffHeapIdentifier on a method declaration

[blocked URL](#)

## 5 @Released

This annotation is used on a parameter declaration to indicate that the method

will call release on the parameter if it is an off-heap reference. When used on method declarations this annotation indicates that the method will call release on one or more field members of the class instance.

### 5.1 Targets

- Method
- Parameter

### 5.2 Retention

- Source

### 5.3 Members

- value - Type is OffHeapIdentifier[]. Optional. Used to match this @Released with one or more off-heap annotations.

### 5.4 Examples

1. @Released on a method parameter

[blocked URL](#)

## 6 @Unretained

This annotation is used on methods to indicate that the return value may be an off-heap reference that the caller is responsible for calling retain on.

### 6.1 Targets

- Method

### 6.2 Retention

- Source

### 6.3 Members

- value - Type is OffHeapIdentifier[]. Optional. Used to match this @Unretained with one or more off-heap annotations.

### 6.4 Examples

1. @Unretained

[blocked URL](#)

## 7 Annotation Matrix

The following matrix summarizes which off-heap annotations are legal for each Java target:

Annotation	Local Variable	Field	Method	Parameter	Constructor
@Offheap	yes	yes	no	yes	no
@Released	no	no	yes	yes	no
@Retained	no	no	yes	yes	yes
@Unretained	no	no	yes	no	no