

# **1 APR::BucketAlloc - Perl API for Bucket Allocation**

## 1.1 Synopsis

```
use APR::BucketAlloc ();
$ba = APR::BucketAlloc->new($pool);
$ba->destroy;
```

## 1.2 Description

APR::BucketAlloc is used for bucket allocation.

### 1.2.1 *new*

Create an APR::BucketAlloc object:

```
$ba = APR::BucketAlloc->new($pool);
```

- **class:** `APR::BucketAlloc`
- **arg1:** `$pool` (APR::Pool object)

The pool used to create this object.

- **ret:** `$ba` (APR::BucketAlloc object)

The new object.

- **since:** 2.0.00

This bucket allocation list (freelist) is used to create new buckets (via APR::Bucket->new) and bucket brigades (via APR::Brigade->new).

You only need to use this method if you aren't running under httpd. If you are running under mod\_perl, you already have a bucket allocation available via `$c->bucket_alloc` and `$bb->bucket_alloc`.

Example:

```
use APR::BucketAlloc ();
use APR::Pool ();
my $ba = APR::BucketAlloc->(APR::Pool->pool);
my $eos_b = APR::Bucket::eos_create($ba);
```

### 1.2.2 *destroy*

Destroy an APR::BucketAlloc object:

```
$ba->destroy;
```

- **arg1: \$ba ( APR::BucketAlloc object )**

The freelist to destroy.

- **ret: no return value**
- **since: 2.0.00**

Once destroyed this object may not be used again.

You need to destroy \$ba **only** if you have created it via `APR::BucketAlloc->new`. If you try to destroy an allocation not created by this method, you will get a segmentation fault.

Moreover normally it is not necessary to destroy allocators, since the pool which created them will destroy them during that pool's cleanup phase.

## 1.3 See Also

mod\_perl 2.0 documentation.

## 1.4 Copyright

mod\_perl 2.0 and its core modules are copyrighted under The Apache Software License, Version 2.0.

## 1.5 Authors

The mod\_perl development team and numerous contributors.



# Table of Contents:

1	APR::BucketAlloc - Perl API for Bucket Allocation	1
1.1	Synopsis	2
1.2	Description	2
1.2.1	new	2
1.2.2	destroy	2
1.3	See Also	3
1.4	Copyright	3
1.5	Authors	3