

## Contents

- [1 Introduction](#)
- [2 Granularity with subject to Array](#)
- [3 Example](#)
- [4 Related Links](#)

## Introduction

**Granularity** is a term that is used in several areas of computing; generally though it refers to the smallest size that can be set or addressed. For example, the granularity of memory determines the smallest amount of data that can be addressed at any one time.

## Granularity with subject to Array

It is important to choose an array granularity consistent with the expected usage pattern of the array. If too small a value is used, an overhead will be incurred for multiple extra allocations when a large number of elements are added to the array. However, if too large a granularity is chosen, the array will waste storage space.

## Example

```
// Here Granularity is set to 10
CDesCArrayFlat* Array = new (ELeave) CDesCArrayFlat(10);
CleanupStack::PushL(Array);
```

If you create an array with a granularity of 10, it means that your array will always have a multiple of 10 elements allocated at any time. Even if you have only one element in your array, memory is already allocated for 10 of them. The reason behind it is that when you add a new element to your array, no memory allocation is needed at that time. If you already have 10 elements in your array and you add one more, the array will be resized to contain 20 elements.

Thus the granularity of an array is quite important in the low-memory mobile application environment. Having a too large granularity will consumes unused memory (which is not a good thing on [Symbian](#)), and having a too low granularity will add extra time processing, as well as maybe unnecessary memory reallocation.

## Related Links

- [Granularity on Wikipedia](#)
- [Granularity on Webopedia](#)