Multidimensional Array [][] vs [,] [duplicate]

Ask Question

396

98

This question already has an answer here:
What are the differences between a multidimensional array and an array of arrays in C#? *9 answers*

```
double[][] ServicePoint = new double[10][9]; // <-- gives an error (1)
double[,] ServicePoint = new double[10,9]; // <-- ok (2)
```

What's their difference? *(1)* yields an error, what's the reason?

And

```
double d = new double[9]
ServicePoint[0] = d;
```

using *(2)* will prompt an error. Why?

c# arrays multidimensional-array

edited Mar 11 '16 at 19:30

Ronny Brendel
3,613 4 29 54

asked Sep 24 '12 at 14:41

william007
5,387 11 46 106

marked as duplicate by nawfal, Mark Hurd, toniedzwiedz, Mark Johnson, nwellnhof Oct 13 '13 at 17:09

This question has been asked before and already has an answer. If those answers do not fully address your question, please ask a new question.

What is your second piece of code supposed to do anyway? It doesn't make any sense. – harold Sep 24 '12 at 14:46

Assign an array of the same size over..is there a way to do this? – william007 Sep 24 '12 at 14:47

what do you mean, something like `double[,] d = new double[9,9];` ? Oh wait I get what you mean, I think. If you mean "is there a way to initialize an array of arrays all at once", then no, you can't do that. – harold Sep 24 '12 at 14:48

2 The first sample (`[][]`) is usually called a 'jagged array' but when you call it an 'array of array' the problem is easier to understand. – Henk Holterman Sep 24 '12 at 14:49

1 For the record: `double d = new double[9];` should be: `double[] d = new double[9];` – Ronnie 'Madolite' Solbakken May 7 '18 at 1:41

5 Answers

437

One is an array of arrays, and one is a 2d array. The former can be jagged, the latter is uniform.

That is, a `double[][]` can validly be:

```
double[][] x = new double[5][];

x[0] = new double[10];
```

```
x[1] = new double[5];
x[2] = new double[3];
x[3] = new double[100];
x[4] = new double[1];
```

Because each entry in the array is a reference to an array of `double`. With a jagged array, you can do an assignment to an array like you want in your second example:

```
x[0] = new double[13];
```

On the second item, because it is a uniform 2d array, you can't assign a 1d array to a row or column, because you must index both the row and column, which gets you down to a single `double`:

```
double[,] ServicePoint = new double[10,9];

ServicePoint[0]... // <-- meaningless, a 2d array can't use just one
```

UPDATE:

To clarify based on your question, the reason your #1 had a syntax error is because you had this:

```
double[][] ServicePoint = new double[10][9];
```

And you can't specify the second index at the time of construction. The key is that ServicePoint is *not* a 2d array, but an 1d array (of arrays) and thus since you are creating a 1d array (of arrays), you specify only one index:

```
double[][] ServicePoint = new double[10][];
```

Then, when you create each item in the array, each of those are also arrays, so *then* you can specify their dimensions (which can be different, hence the term *jagged* array):

```
ServicePoint[0] = new double[13];
ServicePoint[1] = new double[20];
```

Hope that helps!

edited Sep 24 '12 at 14:50

answered Sep 24 '12 at 14:44

James Michael Hare
32.2k 9 60 78

Hi, (1) gives me syntax error, and what's the reason? – william007 Sep 24 '12 at 14:48

3 You can't specify the second range limit at the time of construction, only the first. You specify the second when you allocate each array that is part of the arrays... I'll update and clarify: – James Michael Hare Sep 24 '12 at 14:51

Hi there.Very nice thank you. Valuable information. – mostafakvd Oct 29 '18 at 10:25

Why is the left side considered the "outer" array? – Aaron Franke Dec 6 '18 at 7:00

55

`double[][]` are called **jagged arrays** , The inner dimensions aren't specified in the declaration. Unlike a **rectangular array**, each inner array can be an arbitrary length. Each inner array is implicitly initialized to null rather than an empty array. **Each inner array must be created manually:** Reference [C# 4.0 in nutshell The definitive Reference]

```
for (int i = 0; i < matrix.Length; i++)
{
    matrix[i] = new int [3]; // Create inner array
    for (int j = 0; j < matrix[i].Length; j++)
        matrix[i][j] = i * 3 + j;
}
```

`double[,]` are called `rectangular arrays`, which are declared using commas to separate each dimension. The following piece of code declares a rectangular 3-by-3 two-dimensional array, initializing it with numbers from 0 to 8:

```
int [,] matrix = new int [3, 3];
for (int i = 0; i < matrix.GetLength(0); i++)
    for (int j = 0; j < matrix.GetLength(1); j++)
        matrix [i, j] = i * 3 + j;
```

edited Apr 20 '16 at 20:07

FranciscoBouza
412 4 17

answered Sep 24 '12 at 14:45

Adil
127k 17 172 188

Hi, (1) gives me syntax error, and what's the reason? – william007 Sep 24 '12 at 14:48

2 The example here is very useful to show how to iterate through each level of the arrays using `GetLength(dimension)`. – DeanoMachino Aug 29 '16 at 2:11

`double[,]` is a 2d array (matrix) while `double[][]` is an array of arrays (jagged arrays) and the syntax is:

19

```
double[][] ServicePoint = new double[10][];
```

edited Feb 20 '13 at 19:01

answered Sep 24 '12 at 14:45

Omar

12.3k 6 34 58

Well, it does seem like half an answer. It's not wrong, but it only addresses a few of the issues in the OP. – Servy Sep 24 '12 at 16:19

204

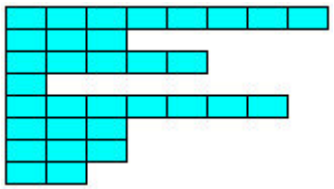
In the first instance you are trying to create what is called a jagged array.

```
double[][] ServicePoint = new double[10][9].
```

The above statement would have worked if it was defined like below.

```
double[][] ServicePoint = new double[10][]
```

what this means is you are creating an array of size 10 ,that can store 10 differently sized arrays inside it.In simple terms an Array of arrays.see the below image,which signifies a jagged array.



http://msdn.microsoft.com/en-us/library/2s05feca(v=vs.80).aspx

The second one is basically a two dimensional array and the syntax is correct and acceptable.

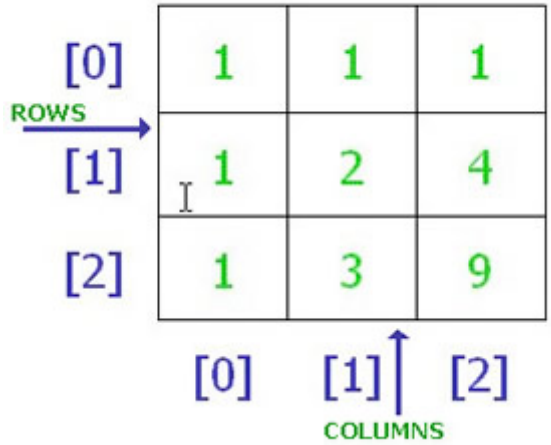
```
double[,] ServicePoint = new double[10,9];//<-ok (2)
```

And to access or modify a two dimensional array you have to pass both the dimensions,but in your case you are passing just a single dimension,thats why the error

Correct usage would be

`ServicePoint[0][2]` ,Refers to an item on the first row ,third column.

Pictorial rep of your two dimensional array



answered Sep 24 '12 at 14:53

Prabhu Murthy
7,746 4 23 32

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

8

`double[][]` is an array of arrays and `double[,]` is a matrix. If you want to initialize an array of array, you will need to do this:

```
double[][] ServicePoint = new double[10][]
for(var i=0;i<ServicePoint.Length;i++)
    ServicePoint[i] = new double[9];
```

Take in account that using arrays of arrays will let you have arrays of different lengths:

```
ServicePoint[0] = new double[10];
ServicePoint[1] = new double[3];
ServicePoint[2] = new double[5];
//and so on...
```

answered Sep 24 '12 at 14:49

Ivo

6,722 4 22 40