

# دورة المحاكاة ببرنامج السميولينك

## SIMULINK

جميع الحقوق محفوظة لمنتديات المهندس

[www.almohandes.org](http://www.almohandes.org)  
[ahmed@almohandes.org](mailto:ahmed@almohandes.org)

2007

# Working with Data Objects

يتيح لنا السميولينك عمل ما يسمى **data objects** وهى التي تستخدم في تحديد قيم المتغيرات فى البرنامج وايضا أنواع تلك المتغيرات ومداها وجميع خصائص البلوكات المستخدمة فى عملية المحاكاة

وكلمة **data** هنا تعنى الإشارات وقيم المتغيرات **signals and parameters**

ويجب معرفة ان هناك أيضا **data classes** والتي تستخدم في تعريف ال **data objects** وتحديد خصائصها

ويمكن أيضا استخدام **data classes** فى تعريف الدوال المختلفة والتي تسمى هنا **methods** .

ويستخدم السميولينك **memory structures** تسمى **packages** وتستخدم فى تخزين البيانات والاكواد اللازمة لتنفيذ ال. **data classes**

وتكون فى السميولينك اسماء الكلاس مثل الشكل الاتى :

**PACKAGE.CLASS**

حيث تحتوى على علامة **dot**

وايضا مثل الاتى :

**Simulink.Parameter**

ف نجد ان الكلاس هو **Parameter** وال **package** الذى ينتمى اليها هى **Simulink**

ويوجد نوعان من الكلاس المستخدم في تعريف الدوال وهما

## Dynamic Methods

فمثلا نفرض ان لدينا `data classe` اسمه `A` ويعرف `method` اسمها `setName` وتستخدم لتعريف اي اسم لاي `instance` في `A` ونفرض ان لدينا `obj` موجود في ال `MATLAB workspace` والان نريد اعادة تسميته

كود:

```
obj.setName('foo');  
setName(obj, 'foo');
```

والطريقة الاخرى هي

## Static Methods

وشكلها كما يلي :

كود:

```
ma = Simulink.ModelAdvisor.getModelAdvisor('vdp')
```

حيث

يستخدم الكلاس `Simulink.ModelAdvisor` والذي يحتوى على الطريقة التي تسمى `getModelAdvisor`

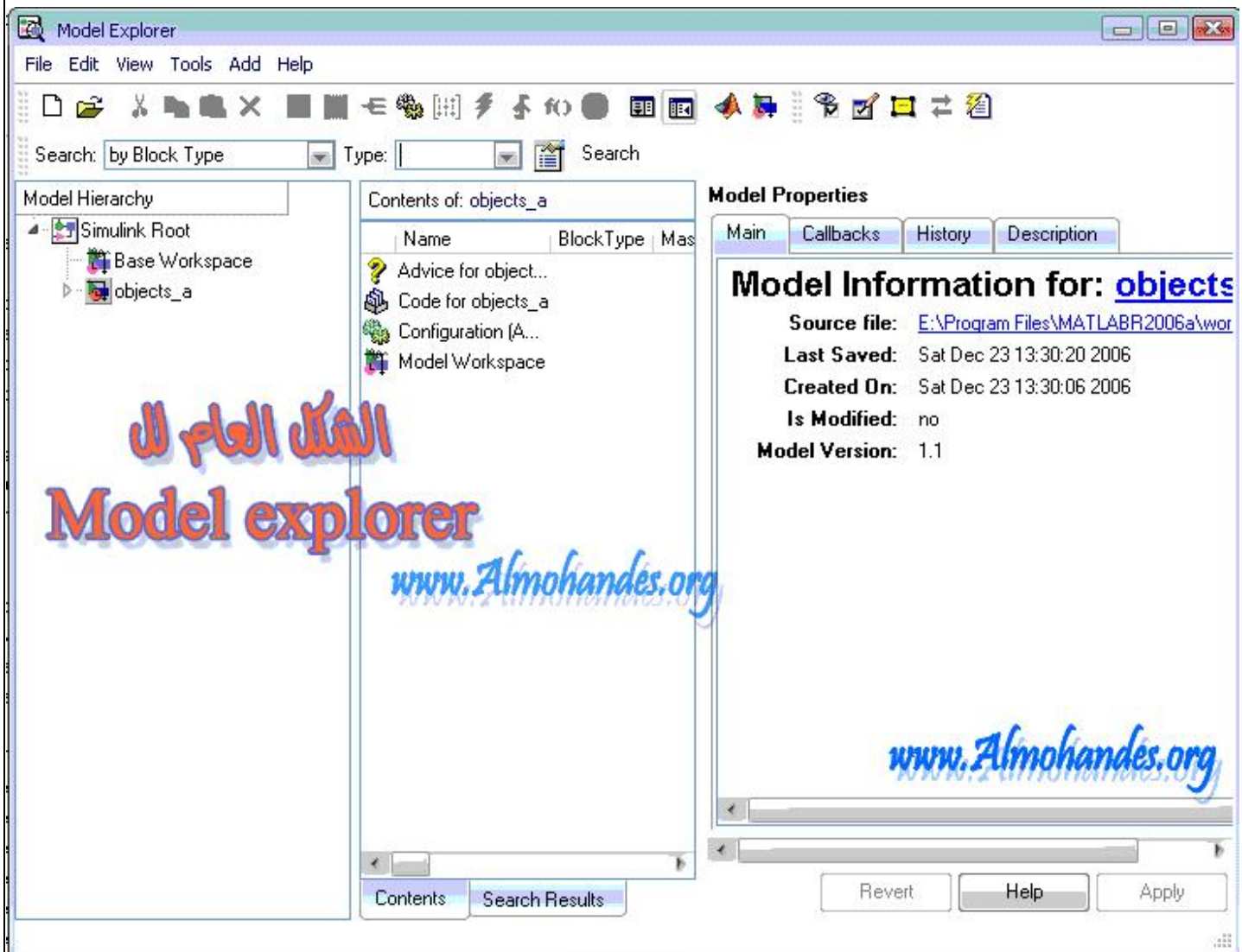
## Constructors

يوجد لكل `data class` طريقة خاصة لإنشاء ال `instances` الخاصة بهذه الكلاس واسم هذه الطريقة مثل اسم الكلاس نفسه

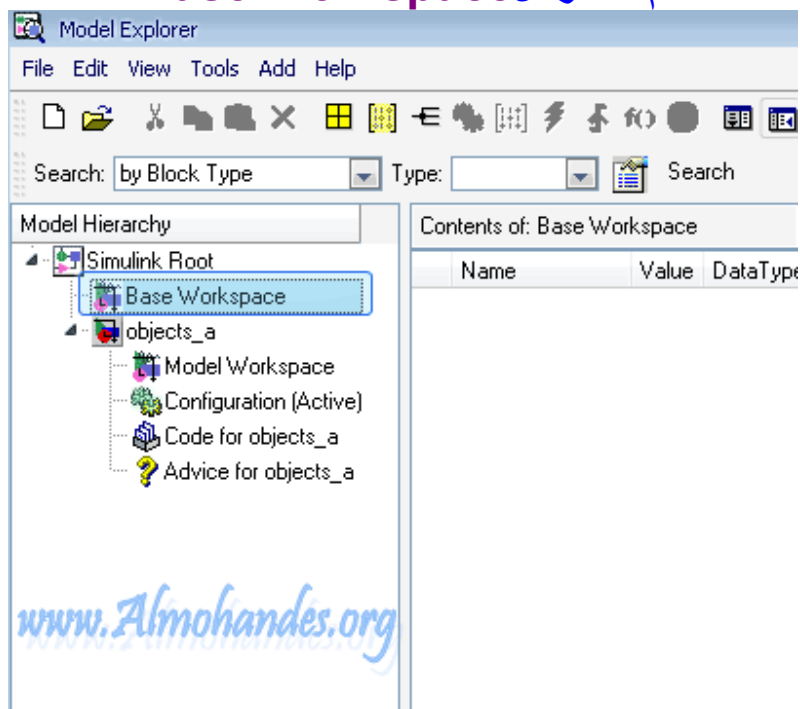
# Using the Model Explorer to Create Data Objects

يمكننا استخدام ال Model Explorer في عمل data objects ويكون عن طريق الخطوات الآتية

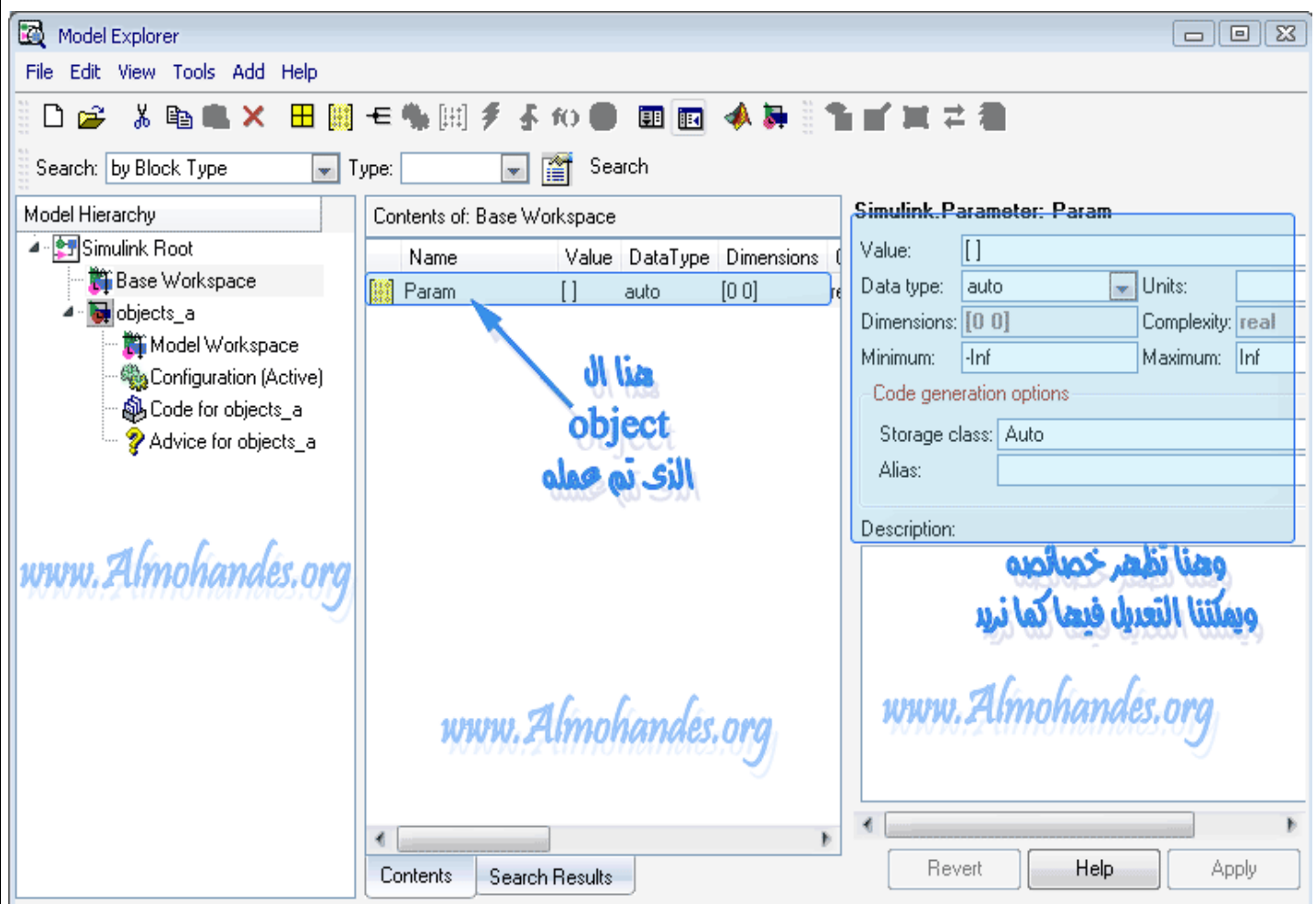
اولا قم بالدخول الى Model Explorer  
من خلال قائمة view



## ثم اختار ال Base Workspace



ثم اختار من قائمة add نوع ال object المراد عمله مثلا  
Simulink Parameter  
Simulink Signal



ويمكننا تعديل خصائص هذا ال object من خلال طريقتين  
الاولى من خلال ال Model Explorer وموضحه في الشكل السابق

**MATLAB commands** والثانية

فمثلا لنفرض ان اسم ال object هو ahmed

فنستخدم الامر في الشكل التالي

```
ahmed.PROPERTY = VALUE;
```

او

```
VALUE = ahmed.PROPERTY;
```

فمثلا نريد وضع القيمة 20 فى القيمة  
نستخدم الامر الاتى

رمز:

```
>> ahmed.value=20
```

ويكون ناتج التنفيذ فى الماتلاب كما يلى :

رمز:

```
ahmed =
```

رمز:

Simulink.Parameter (handle)

RTWInfo: [1x1 Simulink.ParamRTWInfo]

Description: "

DataType: 'auto'

Min: -Inf

Max: Inf

DocUnits: "

Value: 20

Complexity: 'real'

Dimensions: [1 1]

وايضا بالمثل لكل خصائص هذا ال **object**

واذا كنا نريد معرفة خاصية معينة فاننا نقوم بتنفيذ الامر على الصورة الاتية

**ahmed.propertyName**

مثلا

رمز:

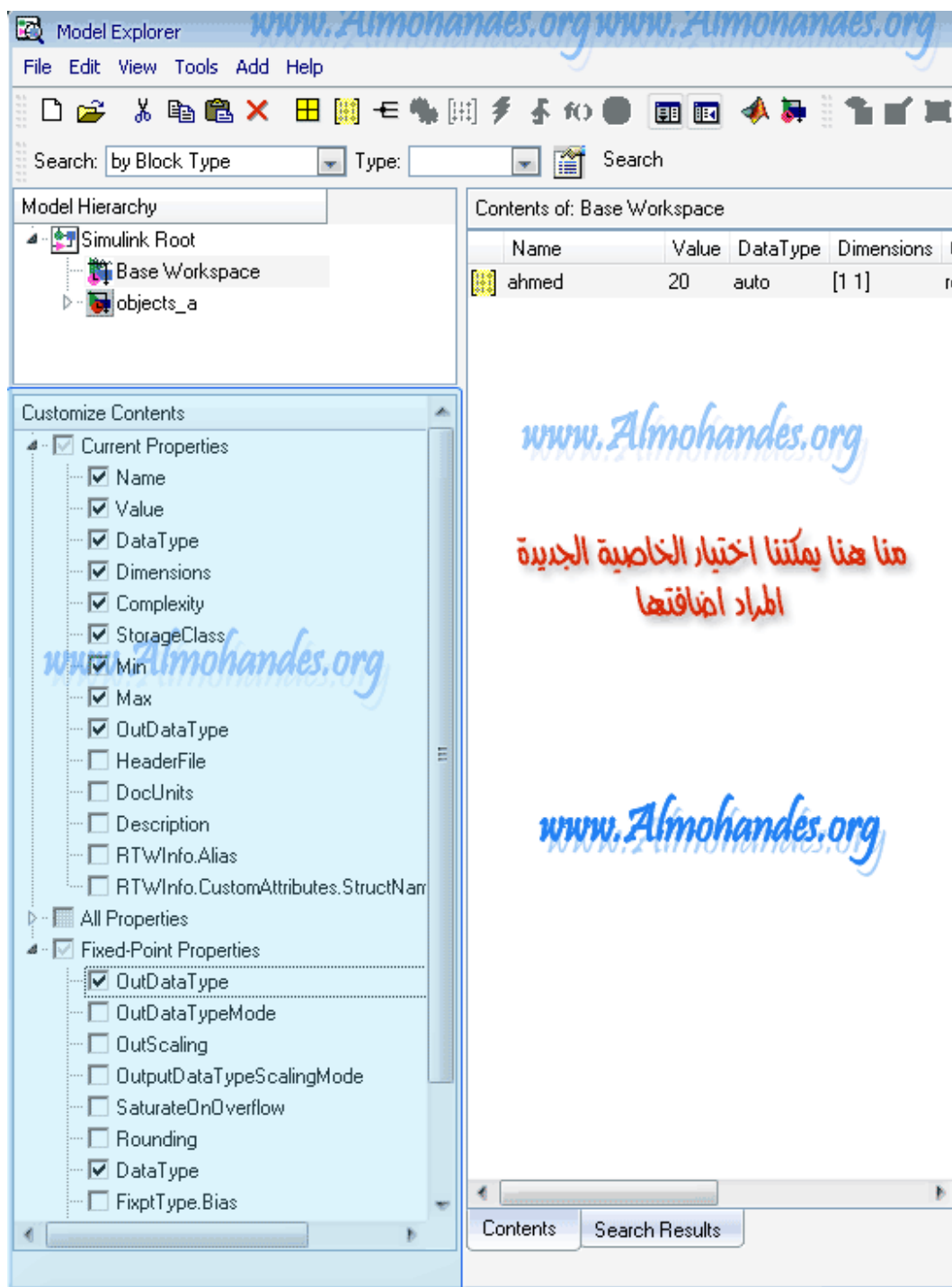
```
>> ahmed.dimensions
```

يرجع لنا الماتلاب ابعاد هذا ال object كما يلي

رمز:

```
ans =  
1    1
```

ويمكننا اضافة خاصية جديدة لل object من خلال  
**Model Explorer**  
كما يلي :



او من خلال اوامر الماتلاب كما يلي  
**اولا** لابد من تعريف الخاصية كما يلي وهذه العملية تسمى **handel**

رمز:

```
>> simulink.bulidtime=10.5
```

اذن الان تم عمل **object** اسمه **simulink** يحتوى على **bulidtime** قيمته تساوى **10.5**

رمز:

```
simulink =  
bulidtime: 10.5000
```

## Handle Versus Value Classes

يوجد نوعان من انواع **data object** في السميولينك وهما  
**value classes**

وتكون عبارة عن **instance** وتحتوى على **MATLAB variable** او **object** من  
النوع **value** مثل

رمز:

```
>> almohandes
```

رمز:

```
almohandes =
```

```
Simulink.AliasType  
Description: 'welcome'  
HeaderFile: ''  
BaseType: 'double'
```

الامر السابق يقوم بعمل **object** اسمه **almohandes**

## handle classes

وهذا يقوم بعمل object من النوع handle ويمكن تخصيص هذا ال handle لأكثر من variable كما فعلنا في المثال السابق

```
22:36 20/07/2017 cannot be indexed using {} or . indexing.
07:12 20/07/2017
02:52 21/07/2017
27:22 27/07/2017
28:00 27/07/2017
16:26 01/08/2017
16:30 01/08/2017
53:50 01/08/2017
30:42 01/08/2017
32:22 01/08/2017
02:18 01/08/2017
03:20 01/08/2017
04:26 01/08/2017
57:54 01/08/2017
15:14 03/08/2017
11:38 13/08/2017
10:00 15/08/2017
05:12 15/08/2017
32:18 21/08/2017
06:42 21/08/2017

>> ahmed

ahmed =

Simulink.Parameter (handle)
    RTWInfo: [1x1 Simulink.ParamRTWInfo]
    Description: ''
    DataType: 'auto'
    Min: -Inf
    Max: Inf
    DocUnits: ''
    Value: []
    Complexity: 'real'
    Dimensions: [0 0]
```

[www.Almohandes.org](http://www.Almohandes.org)

والمعلومية

جميع ال data object classes تكون من النوع value ما عدا **Simulink.Signal** و **Simulink.Parameter**

ويمكننا تحديد النوع سواء كان value او handle عن طريق استعمال سطر او امر الماتلاب كما في المثال الاتي :

رمز:

```
>> gain = Simulink.Parameter
```

فيكون الناتج كما يلي :

رمز:

```
gain =
Simulink.Parameter (handle)
    RTWInfo: [1x1 Simulink.ParamRTWInfo]
    Description: ''
    DocUnits: ''
    Min: -Inf
    Max: Inf
    Value: []
    DataType: 'auto'
    Complexity: 'real'
    Dimensions: [0 0]
```

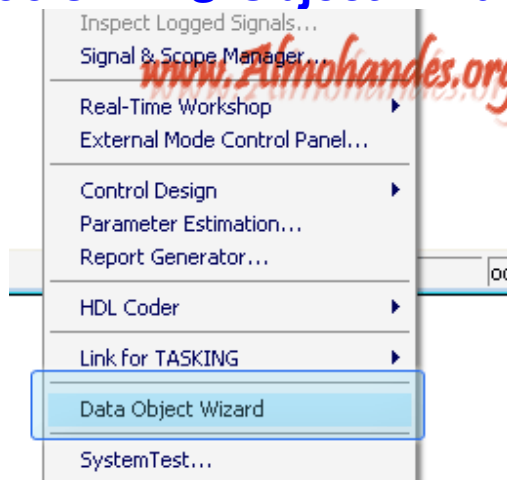
# Saving and Loading Data Objects

يمكننا استخدام امر **save** في حفظ ال **Data Objects** في **MAT-file** ويمكننا ايضا تحميلها في ال **MATLAB workspace** باستخدام الامر **load**

## Data Object Wizard

وهو عبارة عن مجموعة من الخطوات تتيح لنا عمل **data objects** وجعلها مع موديل معين

اولا قم بفتح **Data Object Wizard** من قائمة **tools**



سيفتح لنا الشكل الاتي



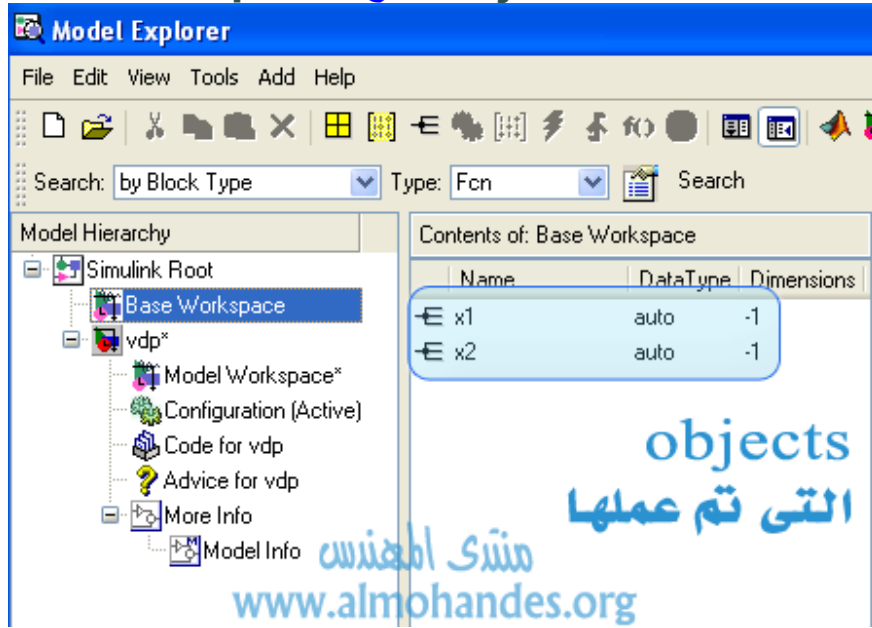
في خانة **Model name** ضع اسم الموديل مثلا ضع الاتي **vdp** وهو موديل موجود في البرنامج لحل معادلة **van der Pol**

ويمكنك ايضا فتح الموديل اولا ثم فتح **Data Object Wizard** وفي هذه الحالة لن تحتاج الى كتابة اى شى فى خانة اسم الموديل لان السميولينك سياخذ اسم الموديل المفتوح وباستخدام **find** يمكننا البحث عن ال **objects** الموجودة فى تحديد بعض الخيارات للبحث وبعد انتهاء عملية البحث ستعرض لنا النتائج فى الجدول العلوى كما فى الصورة الاتية



الان سنقوم باختيار البيانات المطلوب لها عمل **Object** ويكون هذا بوضع علامة صح امام كل نوع فى الجدول العلوى  
 ثم نضغط على **create**

## ولمشاهدة هذه ال objects نفتح Model Explorer



## Data Class Designer

وهذا يتيح لنا عمل Class فرعيه من ال Classes الموجودة فى السميولينك ويقوم ال Data Class Designer بتوليد P-code وهو يحتوى على خصائص الكلاس واسم الكلاس الرئيسى المشتق منه الكلاس الفرعى

## Creating a Data Object Class

من قائمة tool قم بفتح Data Class Designer وسيظهر لنا شكله كما يلى

**Simulink Data Class Designer**

User-defined packages

Package name:

--- None selected ---

New Copy Rename Remove

Parent directory (location of @directory):

Classes Enumerated Property Types Custom Storage Classes

Class name:

Signal New Copy Rename Remove

Derived from: --- None selected --- . --- None selected ---

Properties of this class (inherited properties disabled):

Property Name	Property Type	Factory Value (optional)

New Up Down Remove

Class initialization (optional):

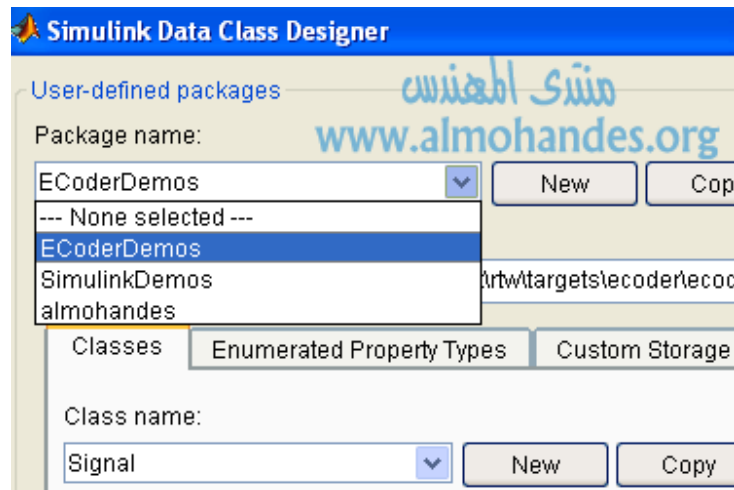
Insert Comments to Assist Writing Class Initialization

www.almohandes.org

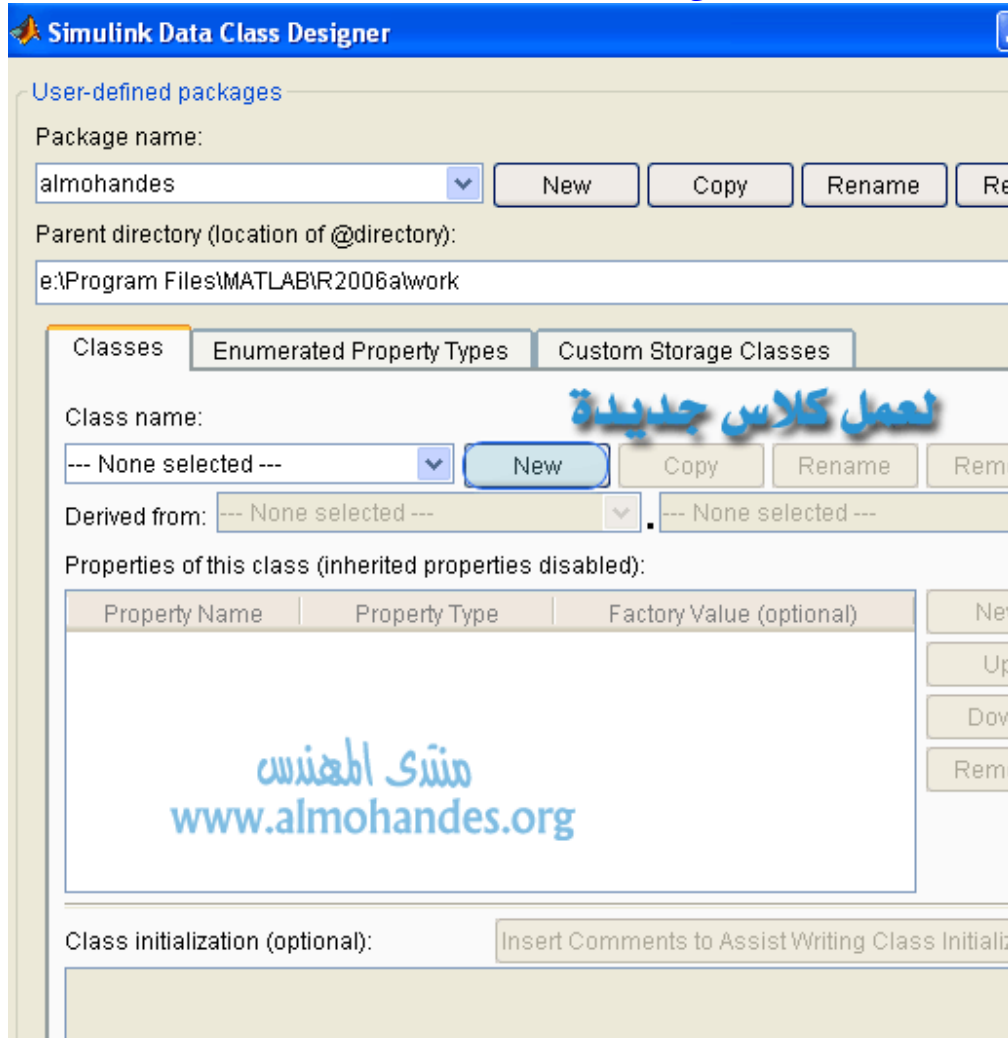
>> Confirm Changes Reload Packages

Help Close

ومن قائمة **package name** سنلاحظ وجود **package** موجود في البرنامج ولكن من الافضل ان نقوم بعمل **package** جديدة ونقوم بعمل الكلاس فيها

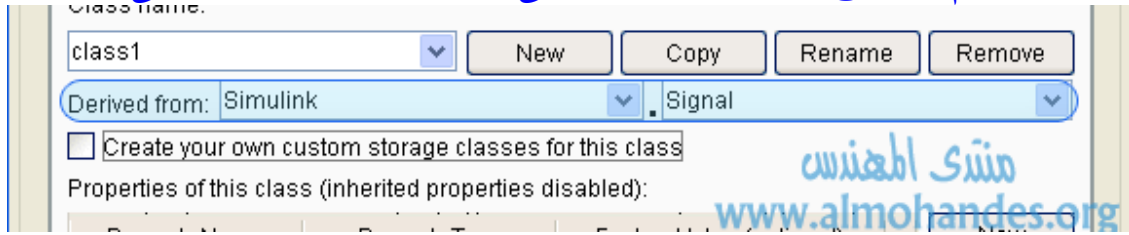


فمثلا سنقوم بعمل package جديدة ويكون اسمها almohandes والان سنذهب الى خانة classes الموجودة في الاسفل ونختار new لعمل كلاس جديدة كما موضح

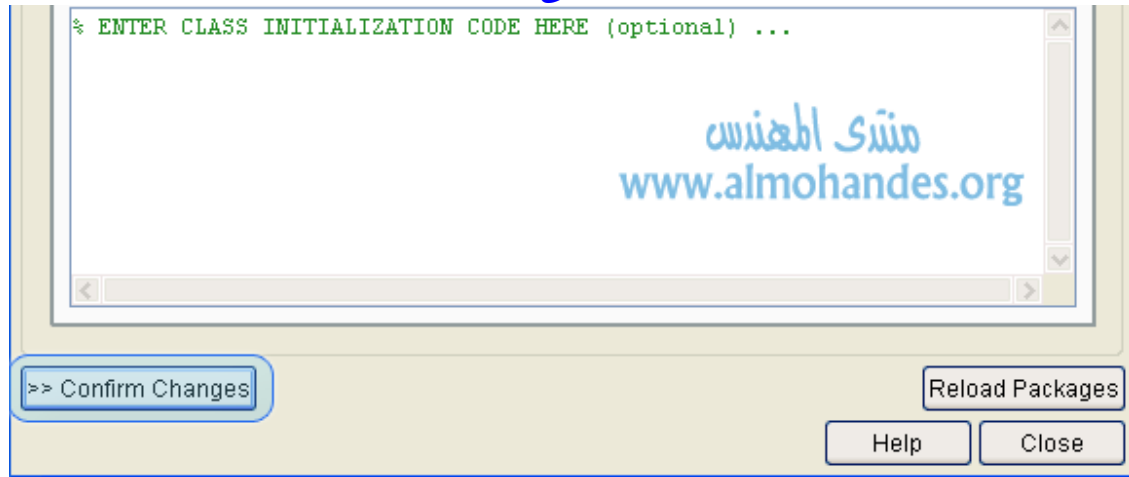


وبعد الضغط على ok سيتم عمل الكلاس الفرعيه والان سنقوم بتحديد الكلاس الاصلى الذى

سيتم اشتقاق الخصائص منه فعلى سبيل المثال اختار كما لى



والان سنقوم بحفظ ال package والتي اسمها **almohandes** والتي تحتوى على كلاس اسمها **class1** وخصائص هذه الكلاس مشتقة من **simulink.signal**  
قم بالضغط على **confirm changes** فى اسفل النافذة  
والذى سينقلك الى النافذة الاتيه



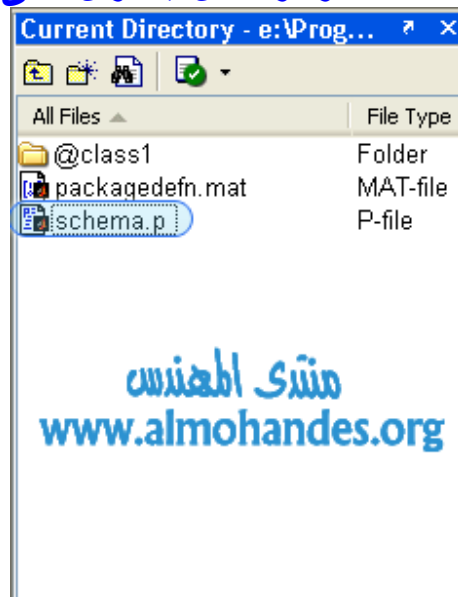
اضغط على **write all** والتي ستقوم بحفظ ال package فى **work directory**



ويمكننا الذهاب الى **work directory** وسنلاحظ وجود ال package الخاصة بنا



وسنلاحظ ايضا وجود **schema.p** وهو الذى يحتوى على الخصائص كما ذكرنا سابقا



ويمكننا الان تعديل في خصائص الكلاس عن طريق الضغط على **new** من على اليمين

Derived from: Simulink Signal

☐ Create your own custom storage classes for this class

Properties of this class (inherited properties disabled):

Property Name	Property Type	Factory Value (optional)
Complexity	--- Unknown ---	
SampleTime	MATLAB array	
SamplingMode	--- Unknown ---	
InitialValue	string	
Prop1	MATLAB array	

New Up Down Remove

www.almohandes.org

وسنلاحظ ايضا ان الخصائص المتوارثه من الكلاس الرئيسى ممنوع التعديل فيها  
والان يمكننا اختيار نوع الخاصيه من القائمة

Properties of this class (inherited properties disabled):

Property Name	Property Type	Factory Value (optional)
Complexity	--- Unknown ---	
SampleTime	MATLAB array	
SamplingMode	--- Unknown ---	
InitialValue	string	
Prop1	MATLAB array	

Class Initialization (optional)

```
% ENTER CLASS INITIALIZATION CODE HERE (optional)
```

double  
int32  
bool  
on/off  
string  
handle

او يمكننا الذهاب الى الخانة التالية التى تسمى **Enumerated Property Types** وهى  
تحتوى على انواع الخصائص التى قمنا بعملها وتظهر ايضا فى القائمة الموجودة فى الصورة  
السابقة

كما فى الصورة الاتية

User-defined packages

Package name:  
 almohandes New Copy Rename Remove

Parent directory (location of @directory):  
 e:\Program Files\MATLAB\R2006a\work

Classes **Enumerated Property Types** Custom Storage Classes

Property type name:  
 ahmed New Copy Rename Remove

Enumerated strings (one per line):

Apply

NOTE:

- Enter one enumerated string per line.
- Do not enclose strings inside single quotes.

قم بالضغط على **new** لعمل نوع جديد ثم قم بتحديد اسمها ونقوم ايضا بعد الضغط على **ok** بتحديد ما يسمى **Enumerated strings** وهي تحتوى على الاحتمالات الممكنة لهذا النوع من الخاصية فمثلا اذا كان النوع هو اللون فقد نجد ان **Enumerated strings** تحتوى على الالوان الممكنة مثل الاحمر والاخضر وغيرها كما موضح

Property type name:  
 colortype New Copy

Enumerated strings (one per line):

red  
 green  
 blue  
 black

ثم قم بالضغط على **apply**  
و **confirm** في الاسفل  
ثم **write all** كما تم عملها سابقا  
والان سنرجع الى خانة الكلاس سنلاحظ وجود الخاصية التي قمنا بعملها

Classes Enumerated Property Types Custom Storage C

Class name:  New Copy

Derived from:  Signal

☐ Create your own custom storage classes for this class

Properties of this class (inherited properties disabled):

Property Name	Property Type	Factory Value (optional)
Complexity	--- Unknown ---	
SampleTime	MATLAB array	
SamplingMode	--- Unknown ---	
InitialValue	string	
Prop1	MATLAB array	
	--- Unknown ---	
	MATLAB array	
	double	
	int32	
	bool	
	on/off	
	string	
	handle	
	colortype	

Class Initialization (optional)

```
% ENTER CLASS INITIALIZATION CODE HERE (optional)
```

انتهى