

Computational BIM Developer Workshop

Rachel Williams

Product Owner
Dynamo

Michael Kirschner

Full Stack Developer
Dynamo

Racel Williams

Product Owner / Senior UX Designer
Dynamo, Autodesk Inc.

M.Arch (Architecture)
M.S. Human-Computer Interaction (CS)

Working on Dynamo for 3 years

Tangible Interaction, Augmented Reality, InfoViz



@racelwilliams

Michael Kirschner

Full Stack Developer

Dynamo, Autodesk Inc.

M.Arch

SMArchS Design Computation

Working on Dynamo for 3 - 4 years

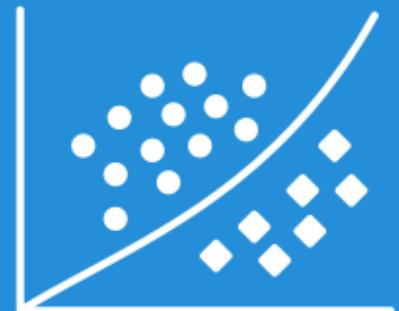
visual programming languages, FPGAs, hardware
hacking, 3d graphics



@MJKKirschner



Who?





Where's the advanced class?



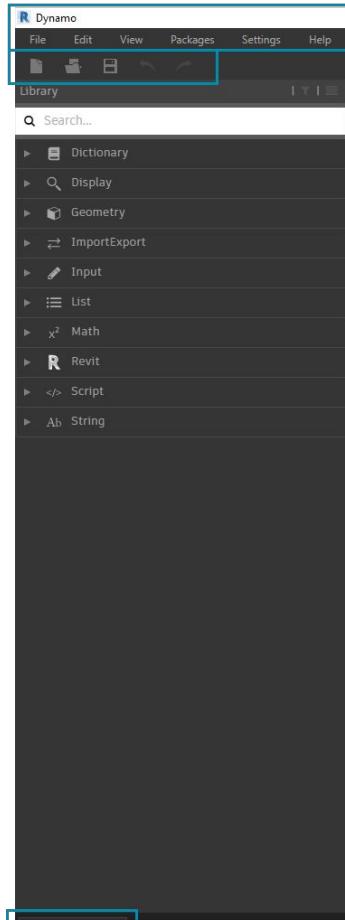
What are we doing here?

1

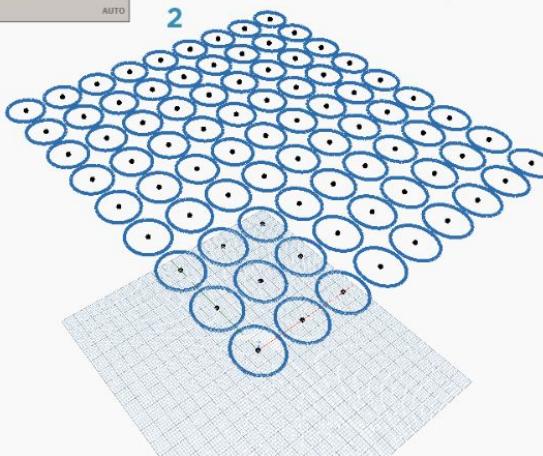
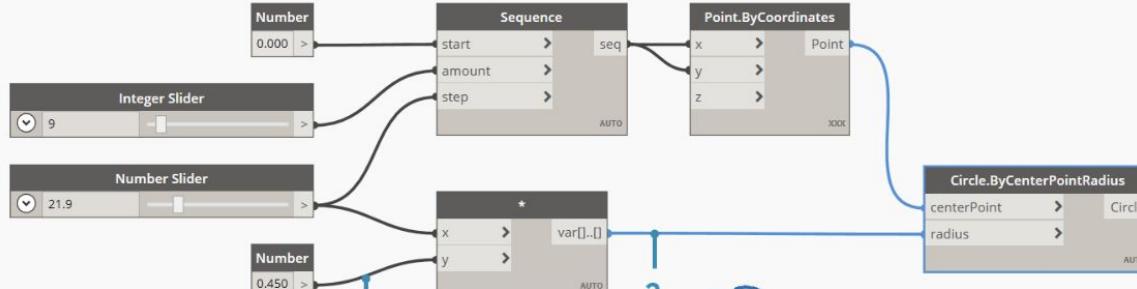
2

3

5



4



1 2 3 4 5

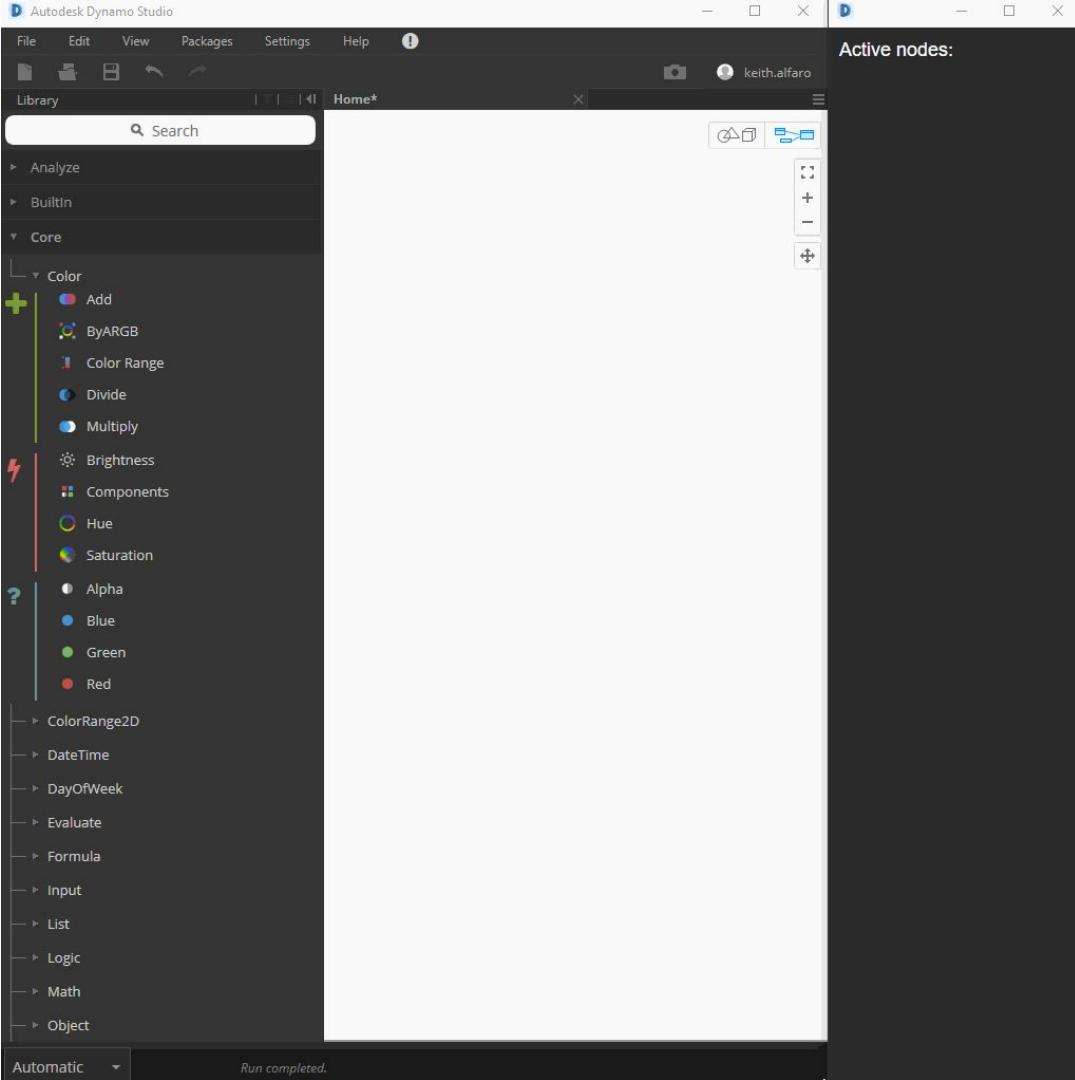


More Functionality

Better Customization

Less Dependency

<https://github.com/DynamoDS/DynamoSamples>





The Schedule

Start Time	End Time	Activity	Location
08:30 am	10:00 am	Developer General Session	San Polo 3402, Level 3 (100 Theater)
10:00 am	10:30 am	AM Break	
10:30 am	12:00 pm	Developer Breakouts <ul style="list-style-type: none">• View Extensions for C# Beginners• Build Custom User Interfaces• Control Dynamo from the Web• Leveraging Speckle in Dynamo• Custom Graphics Visualization	Murano 3302, Level 3 (25 Rounds) Murano 3303, Level 3 (25 Rounds) Murano 3304, Level 3 (25 Rounds) San Polo 3404, Level 3 (25 Rounds) San Polo 3405, Level 3 (25 Rounds)
12:00 pm	01:00 pm	Lunch	Marco Polo 803, Level 1
01:00 pm	03:00 pm	Developer Breakouts <ul style="list-style-type: none">• View Extensions for C# Beginners• Build Custom User Interfaces• Control Dynamo from the Web• Leveraging Speckle in Dynamo• Custom Graphics Visualization	Murano 3302, Level 3 (25 Rounds) Murano 3303, Level 3 (25 Rounds) Murano 3304, Level 3 (25 Rounds) San Polo 3404, Level 3 (25 Rounds) San Polo 3405, Level 3 (25 Rounds)
03:00 pm	03:30 pm	PM Break	Venetian B, Level 2
03:30 pm	05:30 pm	Comp. BIM Forum	Venetian B, Level 2 (400 Theater)
05:30 pm	06:30 pm	Comp. BIM Forum Beer Bust	Venetian B, Level 2

Forum

Venetian B, Level 2

3:30 pm

Wednesday Nov 14 10pm – 1am



Zach Kron
@ZachKron

Following

Replies to @gurump @LilliMSmith and 5 others

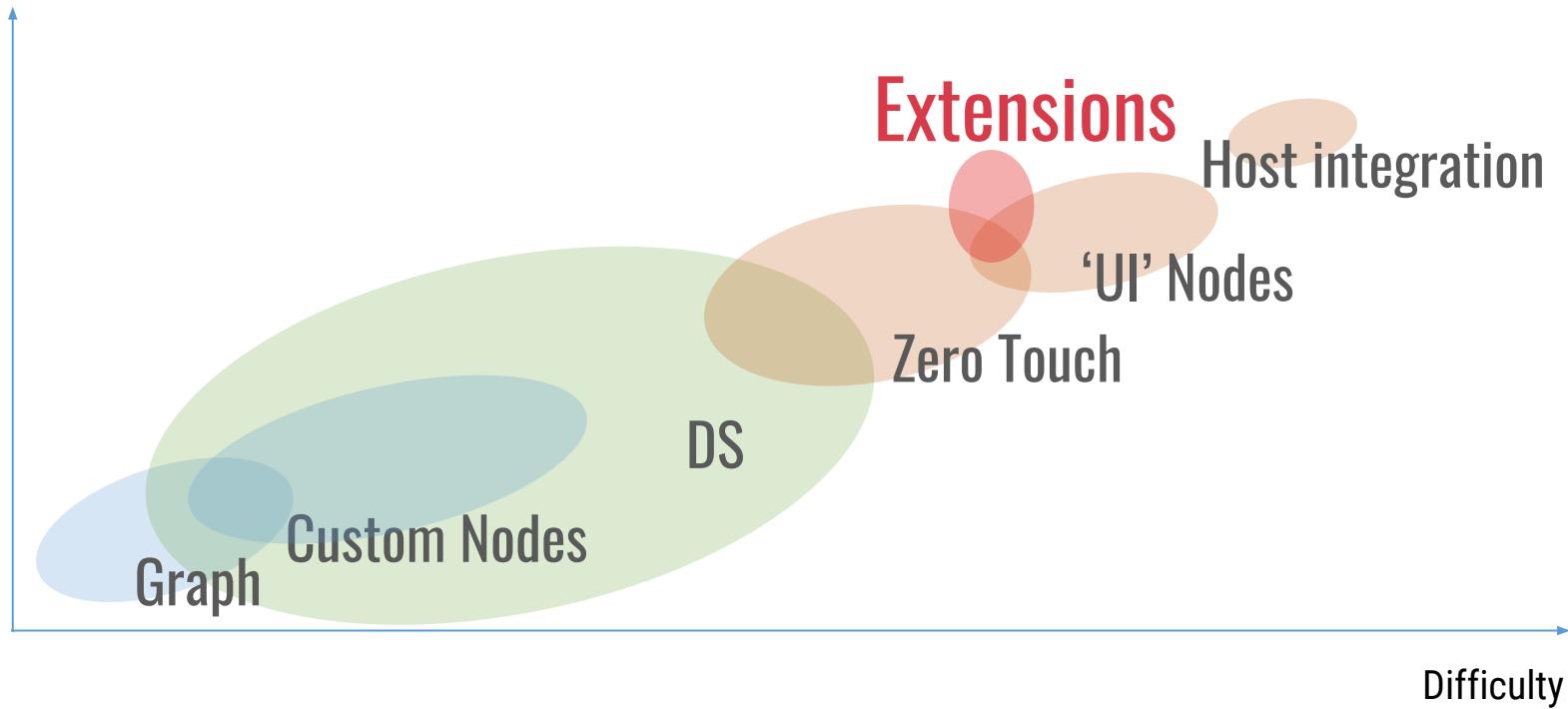
all you have to do is show up and accurately
sketch the outcome of this on a napkin:
 $a = (0..4..#3);$

NurbsCurve.ByControlPoints((List.Flatten(Poi
nt.ByCoordinates(a<1>, a<2>, a<3>), -1)));

6:08 PM - 30 Oct 2018

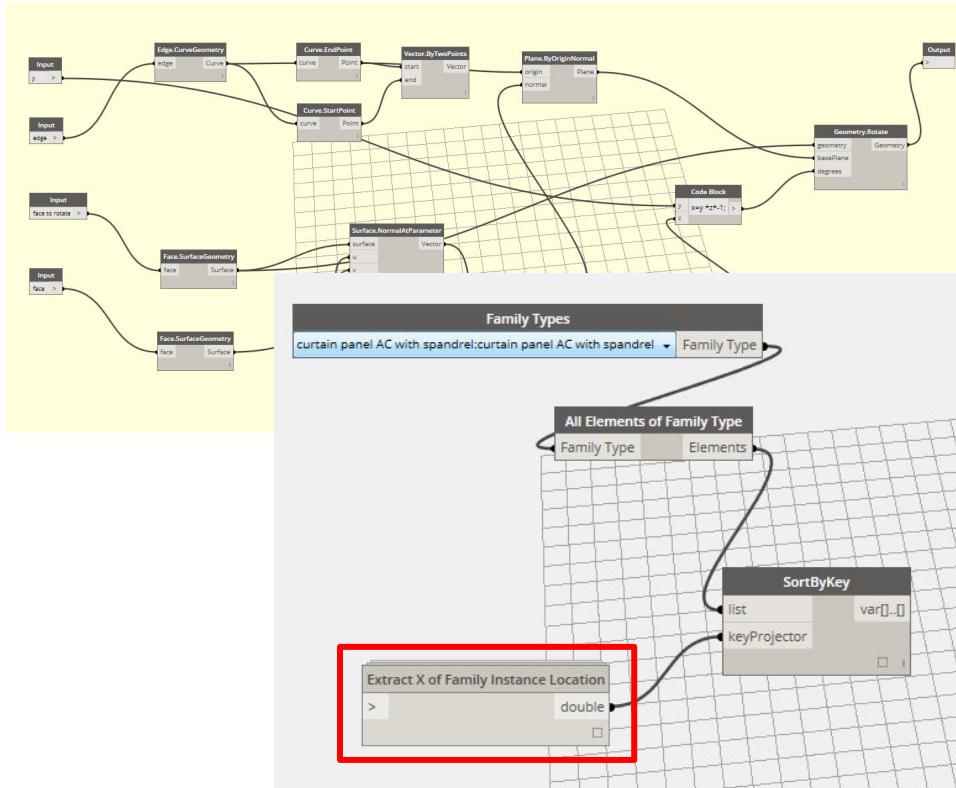


Expressive
Power

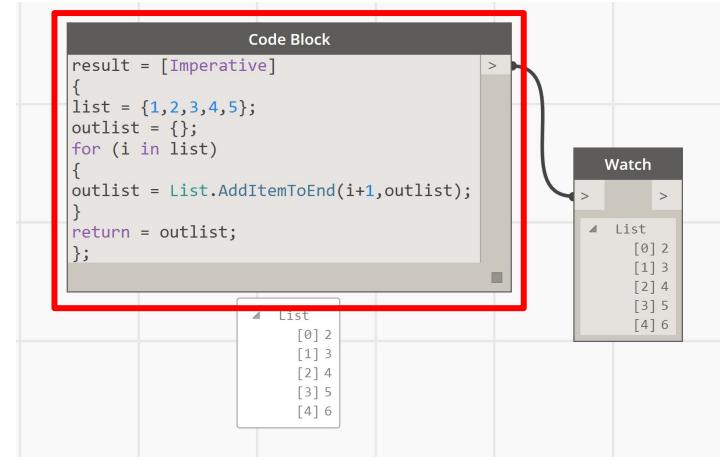


Extending Dynamo

custom nodes



designscript

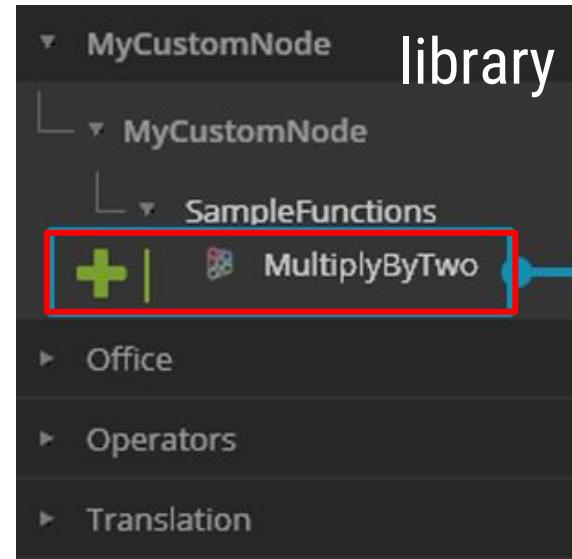


Out-of-the-box

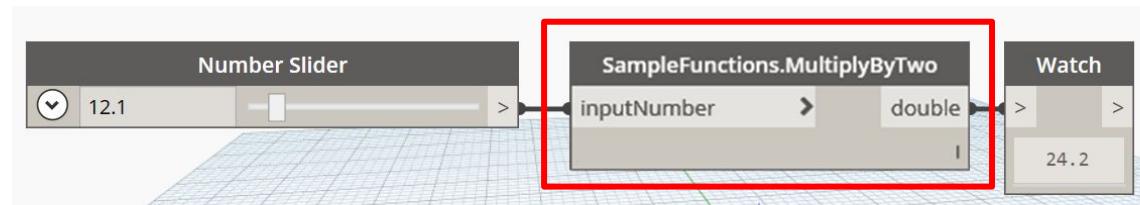
```
namespace MyCustomNode
{
    public class SampleFunctions
    {
        //The empty private constructor.
        //This will be not imported into Dynamo.
        private SampleFunctions() { }

        //The public multiplication method.
        //This will be imported into Dynamo.
        public static double MultiplyByTwo(double inputNumber)
        {
            return inputNumber * 2.0;
        }
    }
}
```

.NET

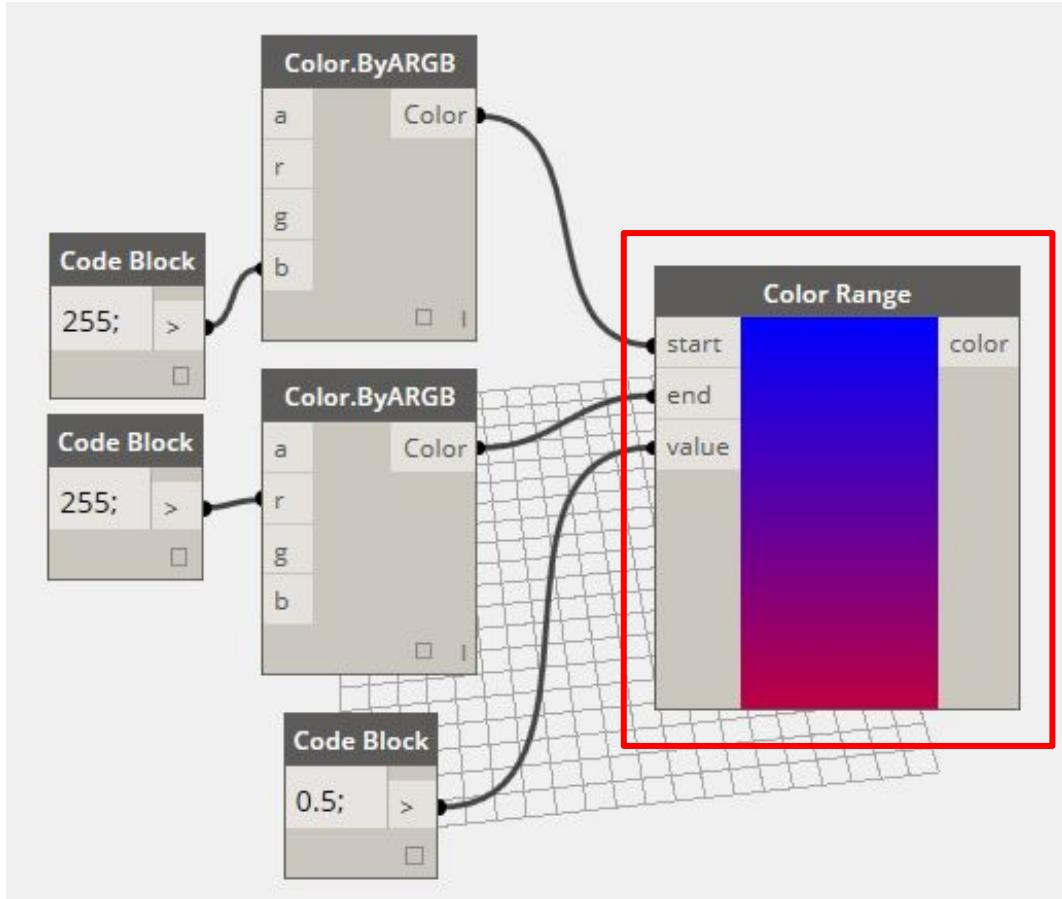


library



Zero Touch

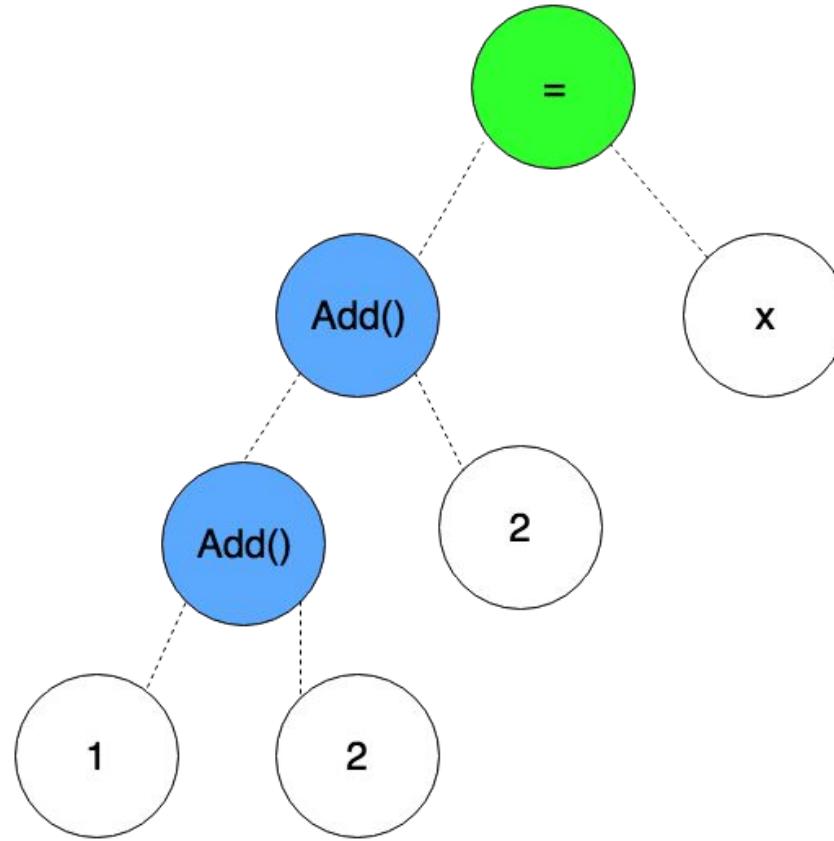
node



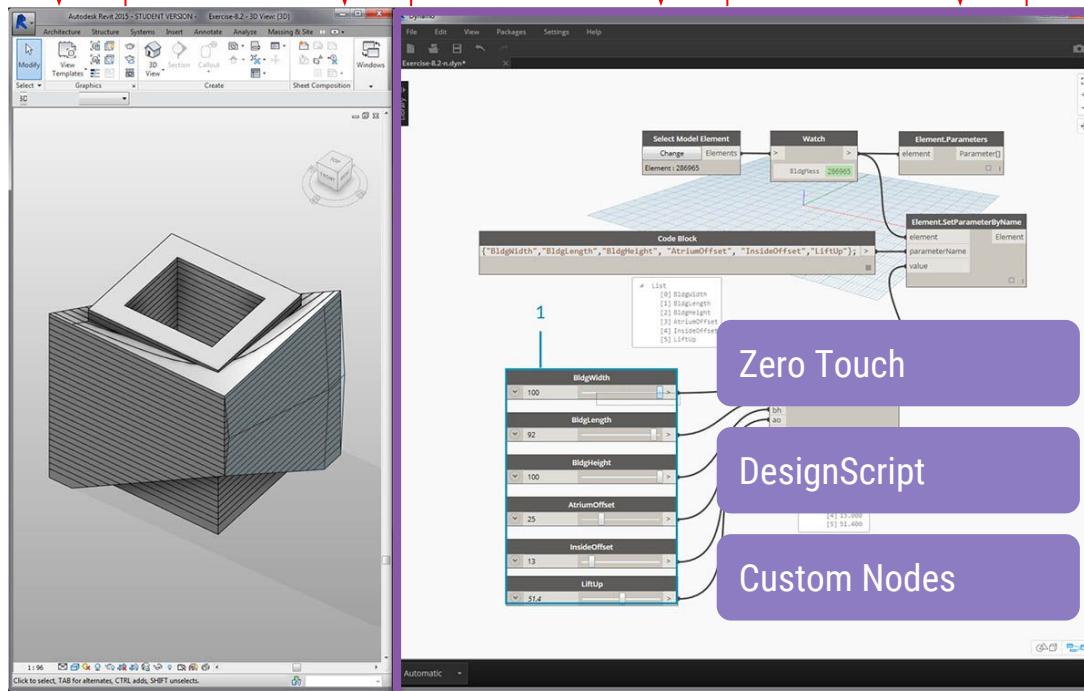
UI Nodes

NodeView Customization
provides WPF UI customization

NodeModel
Defines inputs, outputs, AST



UI Nodes - AST creation



Extensions

Object-Oriented Programming Fundamentals

What the heck is a static class?

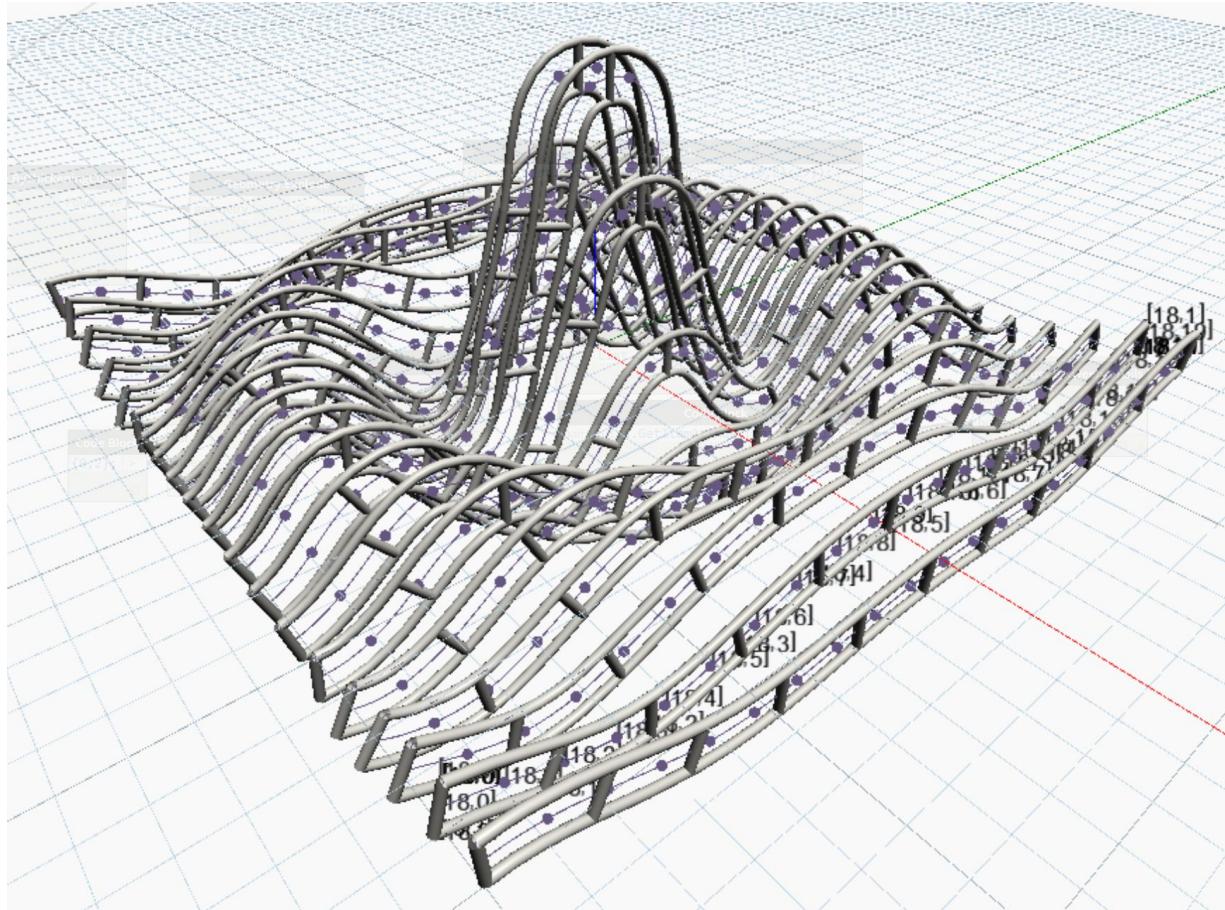
object some data and functions that are stuck together

type what kind of object are you?

class an abstract definition of an object type

instance an object that exists! We can point and say that one!

static a class or function we can't make instances of...
there can only be one!



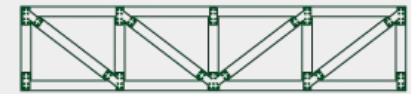
Truss object as an example of OOP

Truss class

```
{  
    bottomPoints,  
    topPoints,  
    generateGeo()  
}
```

Truss instance

new Type



class, instance, type

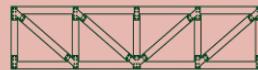
Truss class

```
{  
    bottomPoints,  
    topPoints,  
    generateGeo()  
}
```

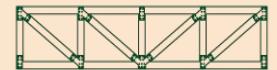
new Type

class, instance, type

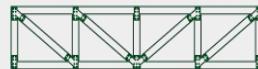
Truss
instance



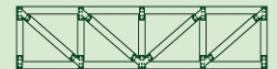
Truss
instance



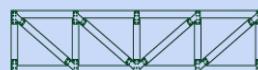
Truss
instance



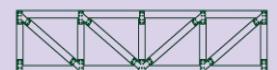
Truss
instance



Truss
instance



Truss
instance



structured data

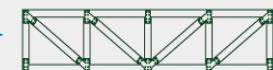
```
▲ List
  ▲ [0] List
    [0] Point(X = -10.000, Y = -1)
    [1] Point(X = -10.000, Y = -5)
  ▲ [1] List
    [0] Point(X = -10.000, Y = -7)
    [1] Point(X = -10.000, Y = -8)
  ▲ [2] List
    [0] Point(X = -10.000, Y = -5)
    [1] Point(X = -10.000, Y = -6)
  ▲ [3] List
    [0] Point(X = -10.000, Y = -4)
    [1] Point(X = -10.000, Y = -4)
  ▲ [4] List
    [0] Point(X = -10.000, Y = -2)
    [1] Point(X = -10.000, Y = -1)
  ▾ [5] List
```

Truss class

field/property
bottomPoints,
topPoints

function
generateGeo()

Truss object

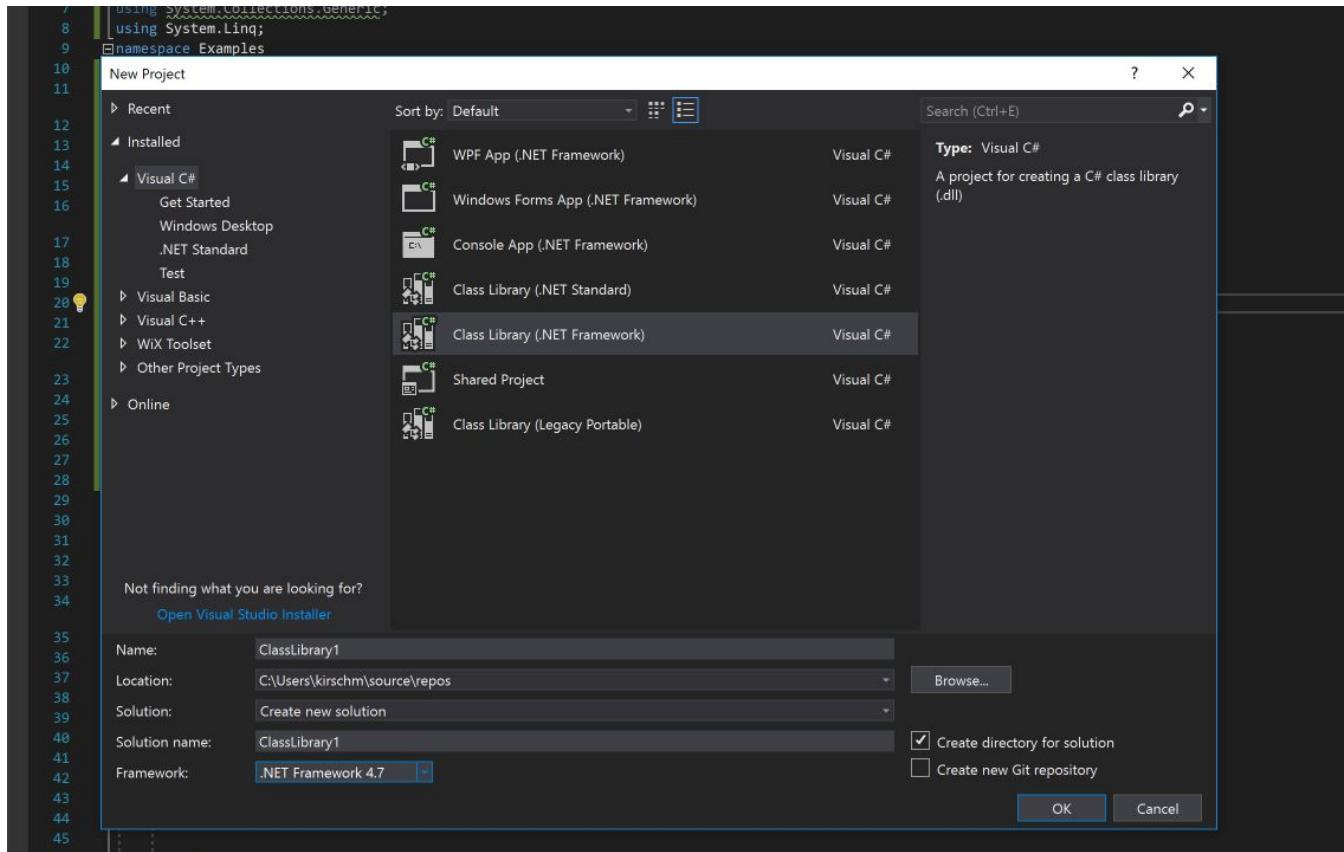


objects bind data and function together

Getting Started with C#

Visual Studio

Get Visual Studio. <https://visualstudio.microsoft.com/>

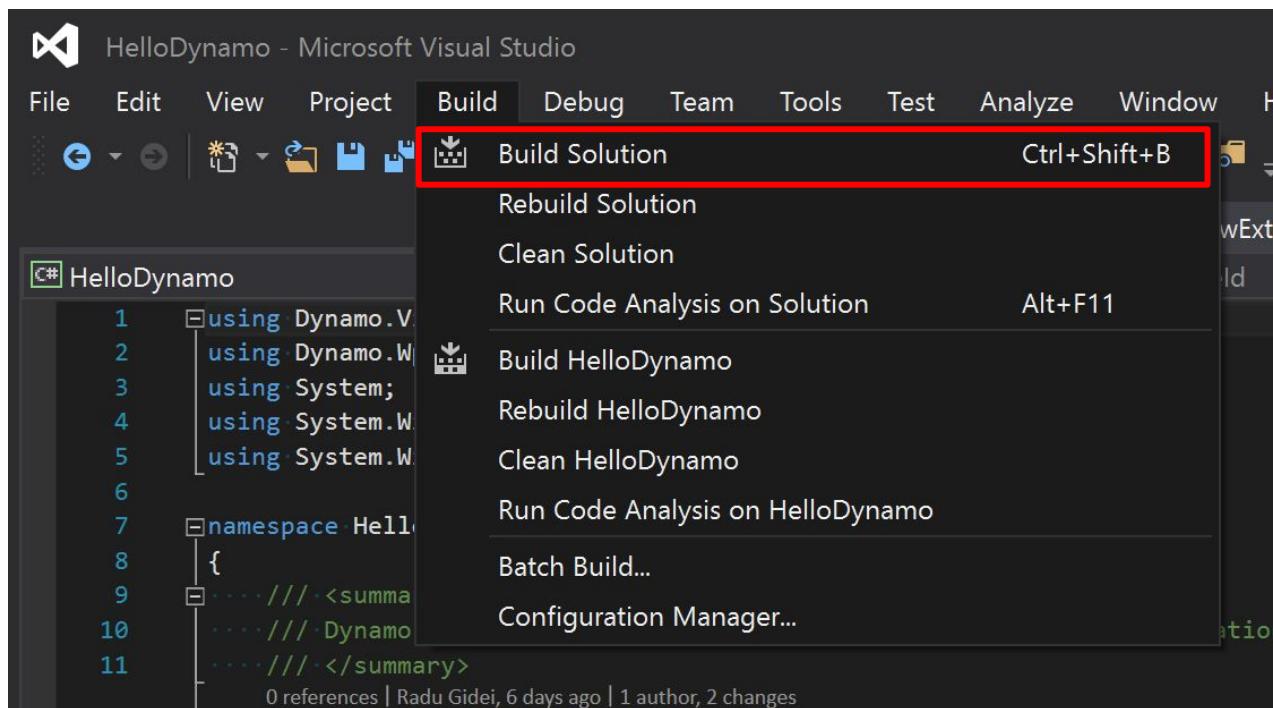


Build Solution

compile files + components

create assemblies (.dll)
create executables (.exe)

detect syntax & type errors



Show output from: Build

```
1>----- Build started: Project: HelloDynamo, Configuration: Debug Any CPU -----
1> HelloDynamo -> C:\GitHub\DeveloperWorkshop\HelloDynamo\Source\HelloDynamo\bin\Debug\HelloDynamo.dll
1> ++++++
1> Started building the Dynamo extension
1> ++++++
1> Built to C:\GitHub\DeveloperWorkshop\HelloDynamo\Source\HelloDynamo\bin\Debug\HelloDynamo.dll
1> ++++++
1> Deployed to C:\Users\williar\AppData\Roaming\Dynamo\Dynamic Core\2.0\packages\HelloDynamo
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

SUCCESS?!

.dll
created



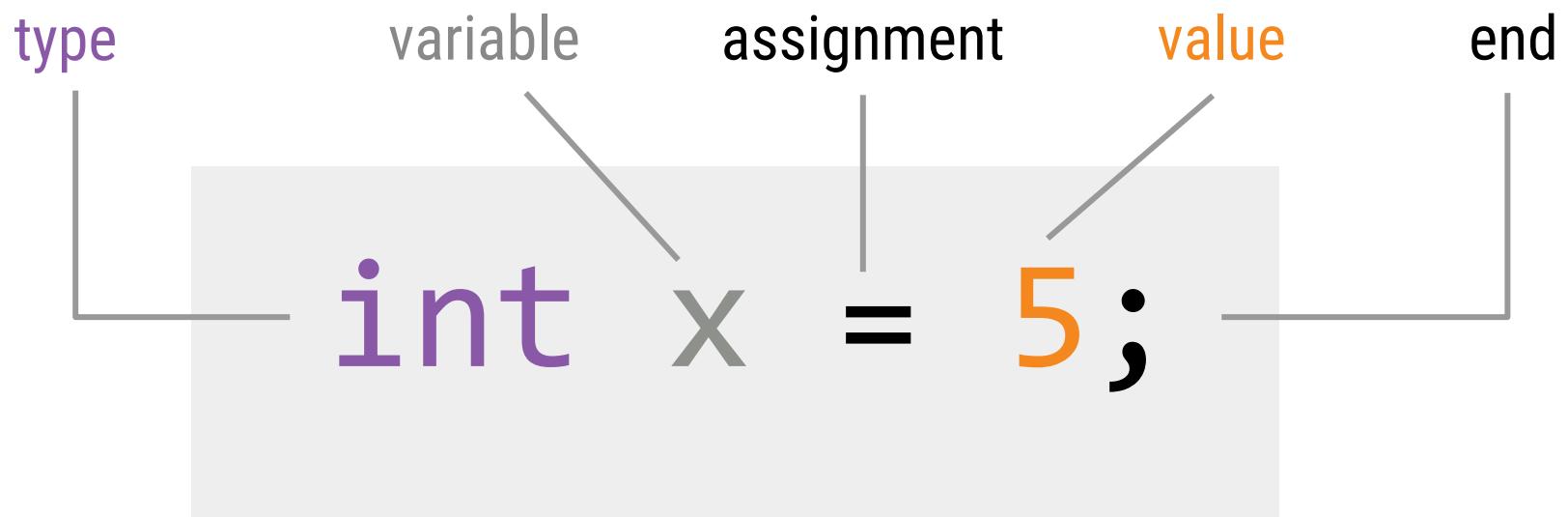
Getting Started with C#

Fundamentals

base types

```
int x = 5; ----- whole number  
double y = 5.1; ----- decimal places  
string hi = "hello"; ----- text  
boolean state = true; ----- true or false
```

statements



functions/methods

modifier
access modifier | return type | method name | parameters

```
public static double AddTwoNumbers(double x, double y)
{
    return x + y;
}
```

The diagram illustrates the structure of a Java method definition. It features a horizontal line with four vertical labels pointing to specific parts of the code: 'access modifier' points to 'public static', 'return type' points to 'double', 'method name' points to 'AddTwoNumbers', and 'parameters' points to 'x' and 'y'. The code itself is displayed below the line.

statically typed languages

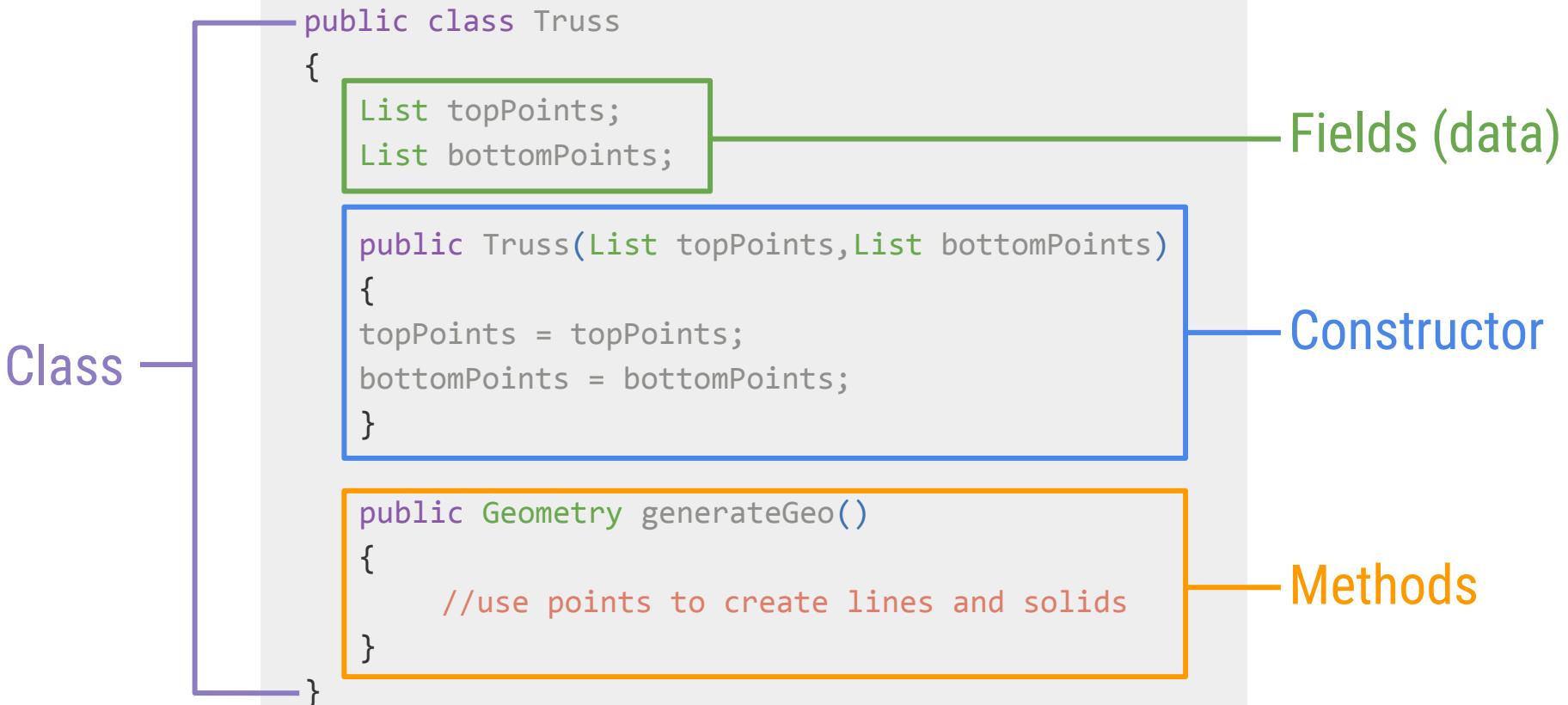
class method Invocation
| | (and arguments)

var number = Math.AddTwoNumbers("hello", 5);



Cannot convert *string* to *double*!

classes & constructors



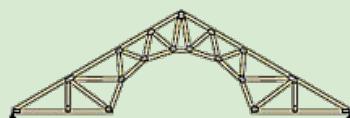
making new instances

```
var truss1 = new Truss(pts1, pts2);  
var truss2 = new Truss(pts3, pts4);  
var truss3 = new Truss(pts5, pts6);
```

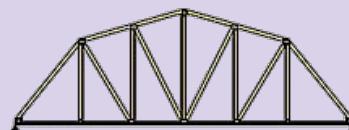
Truss
instance



Truss
instance



Truss
instance

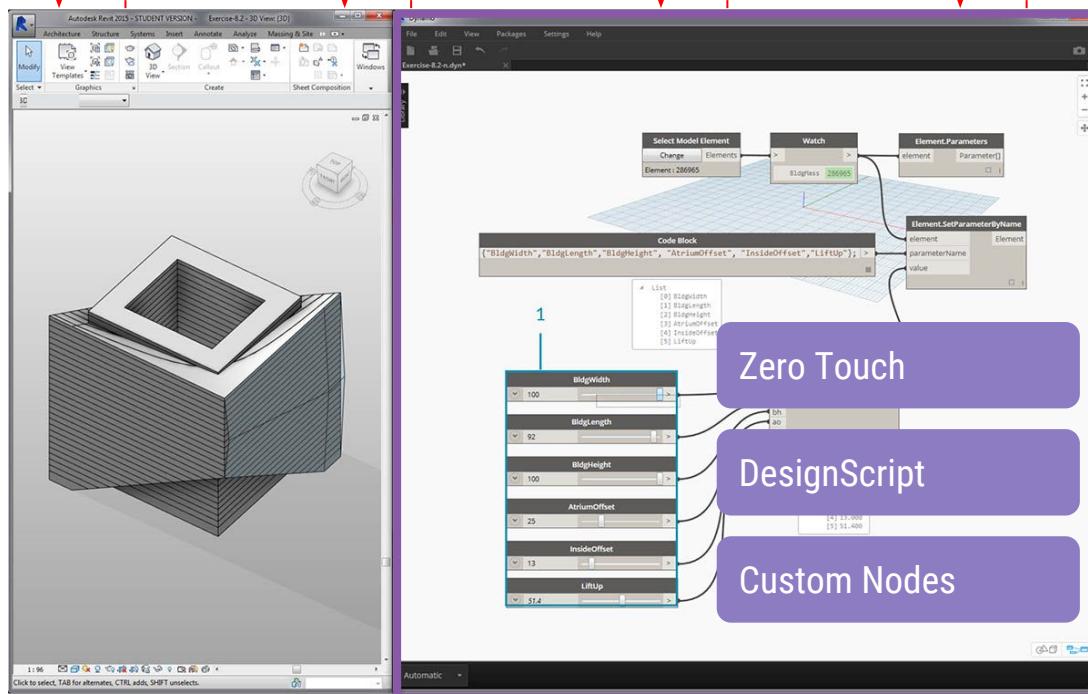


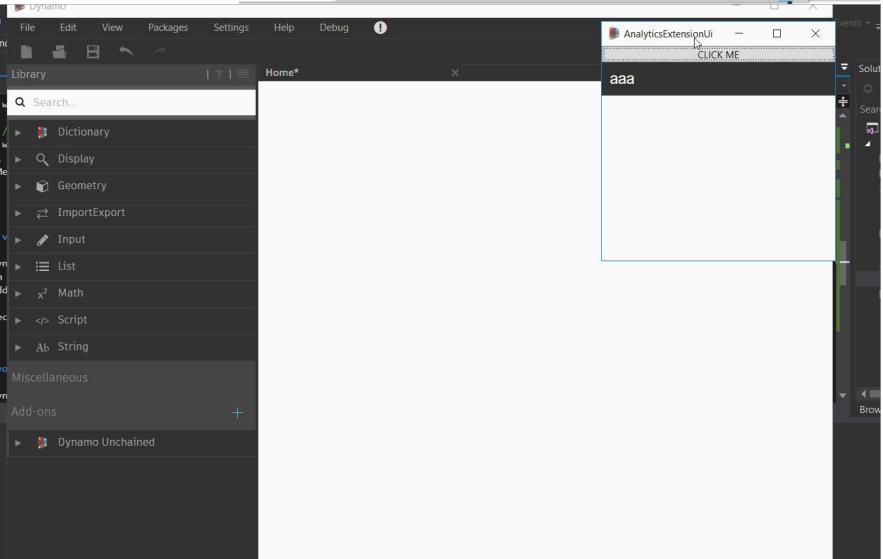
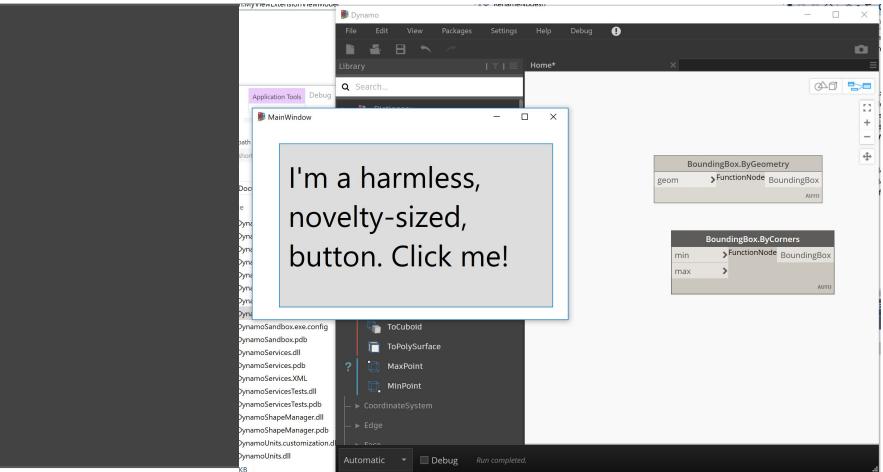
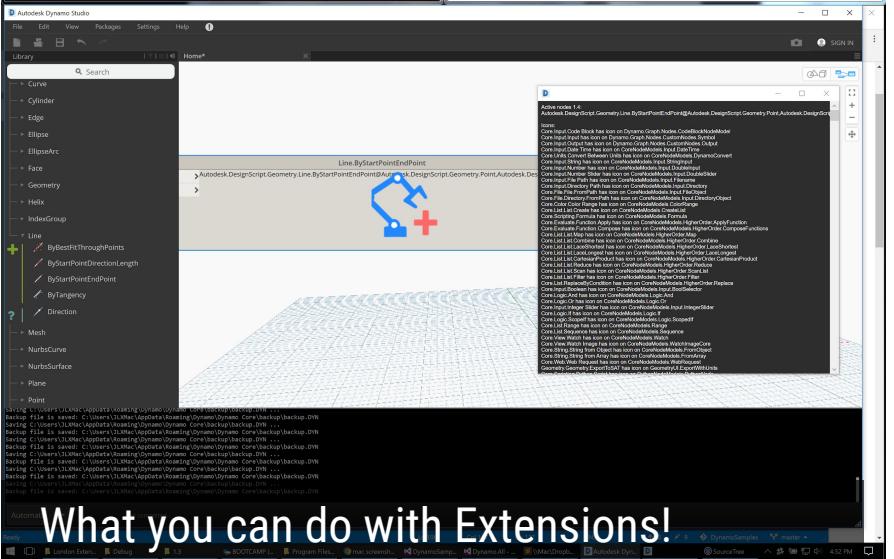
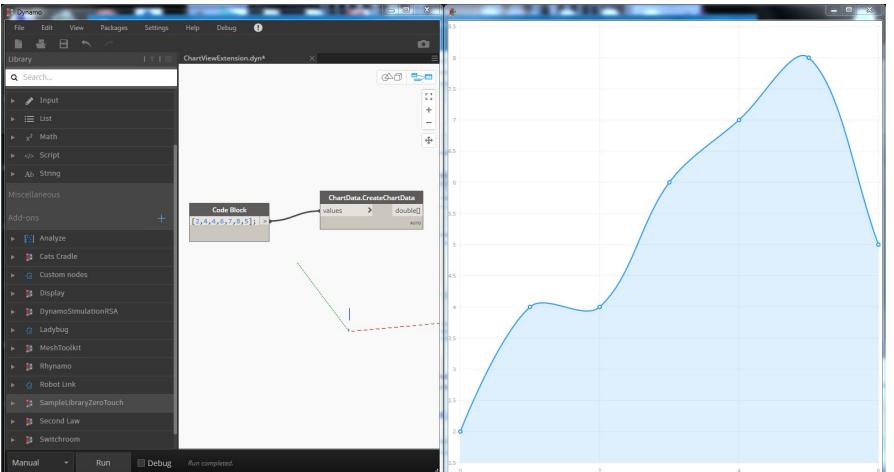
be in the know...

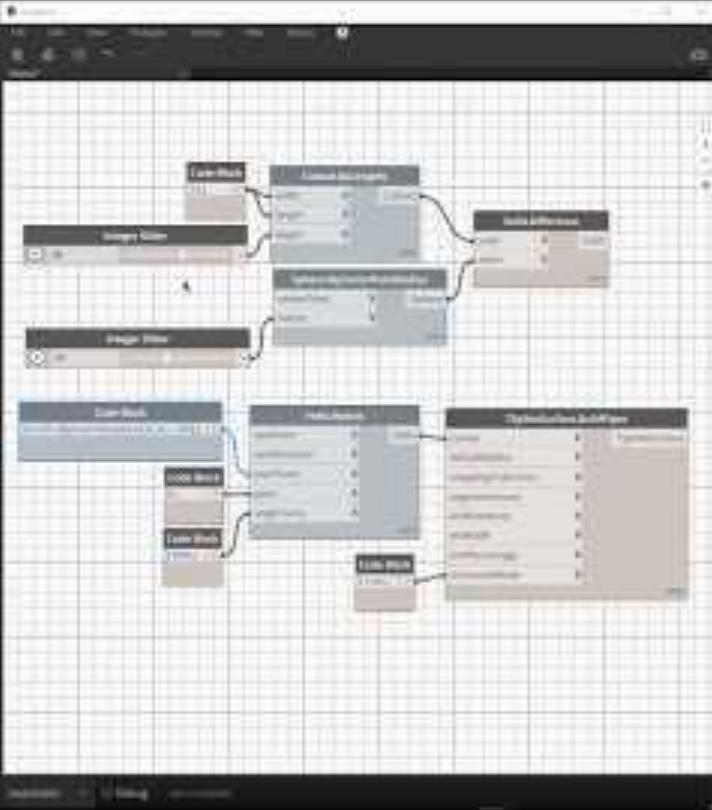
- repo/git
- source code
- Build
- .DLL
- .Net
- WPF
- UIControl
- FrameWorkElement
- Binding
- Event



Back to Extensions!!!







Viewport - Dynamo View Extension

Anatomy of an Extension

extension architecture

1

MyExtension.dll

```
public class MyViewExtension:IViewExtension{}
```

Type that implements
IViewExtension or IExtension

2

MyExtension_ViewExtensionDefinition.xml

```
<ViewExtensionDefinition>
    <AssemblyPath>..\MyExtension.dll</AssemblyPath>
    <TypeName>MyExtensions.MyViewExtension</AssemblyPath>
</ViewExtensionDefinition>
```

XML Manifest

Dynamo/Extensions

MyExtension_ExtensionDefinition.xml

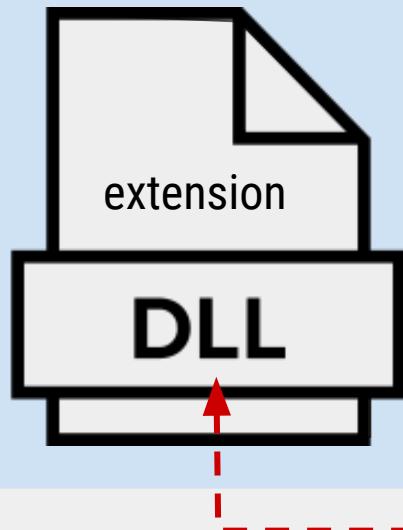
Dynamo/ViewExtensions

MyExtension_ViewExtensionDefinition.xml

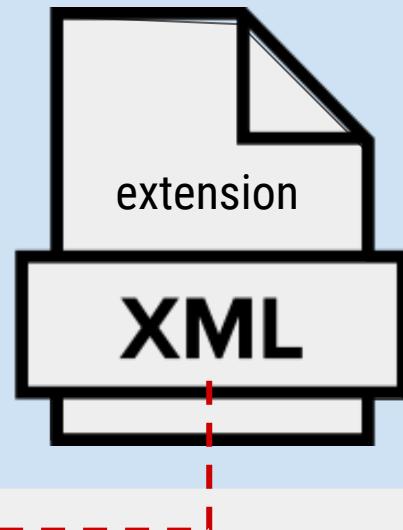
load extensions from Dynamo Core folder

Package Folder

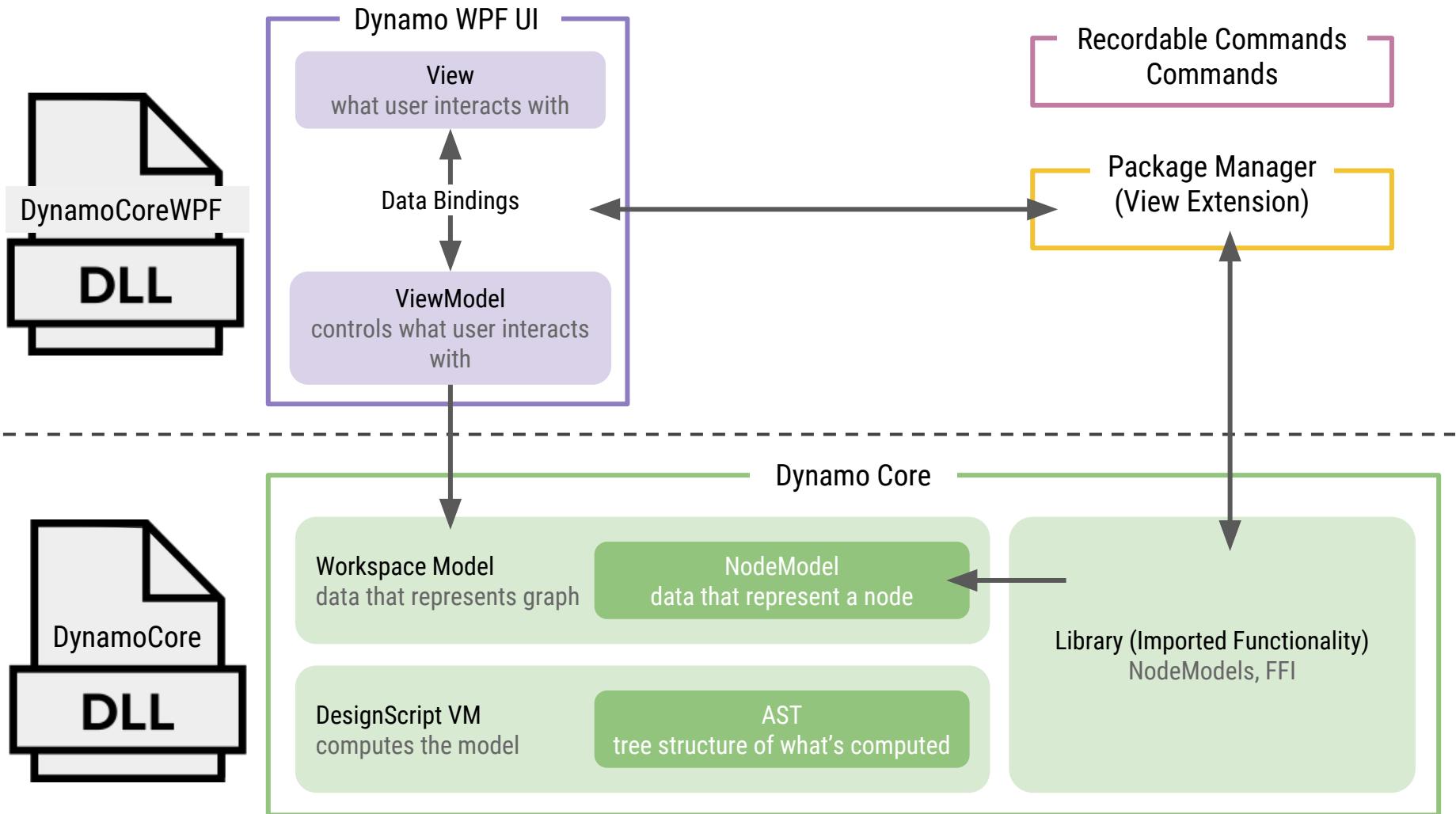
/bin



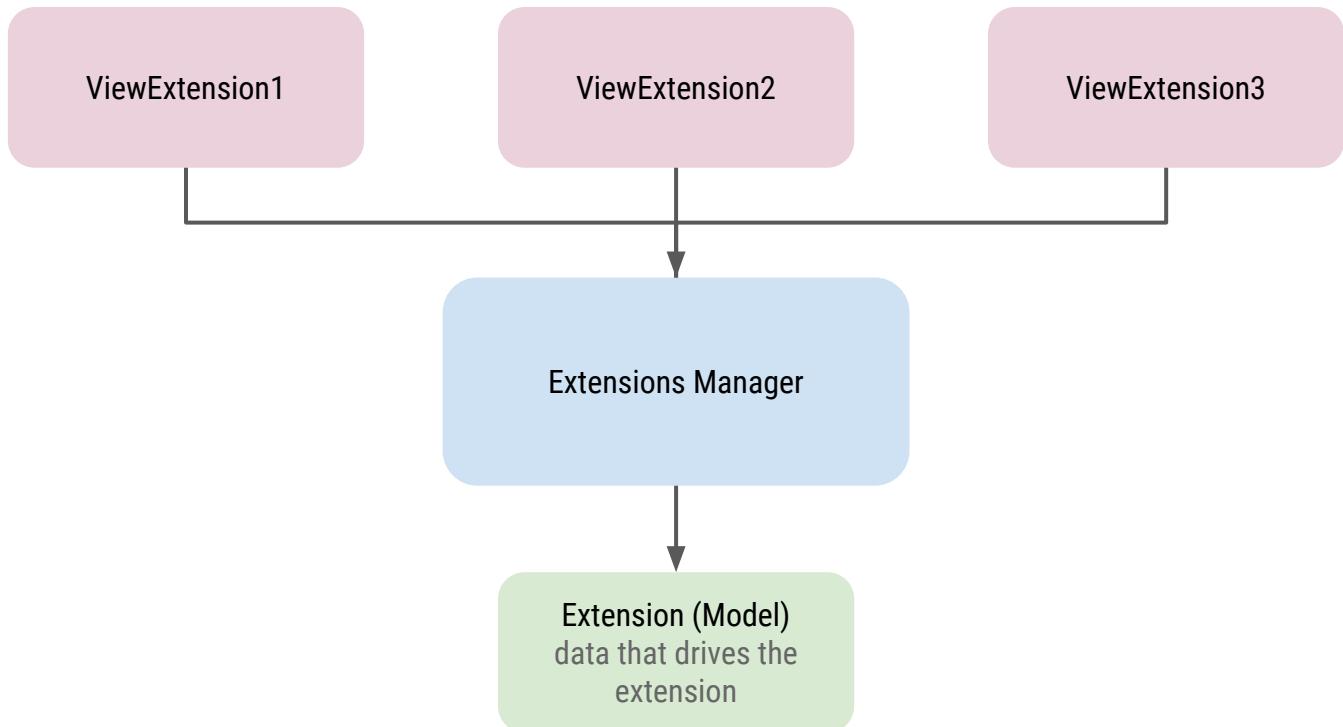
/extra

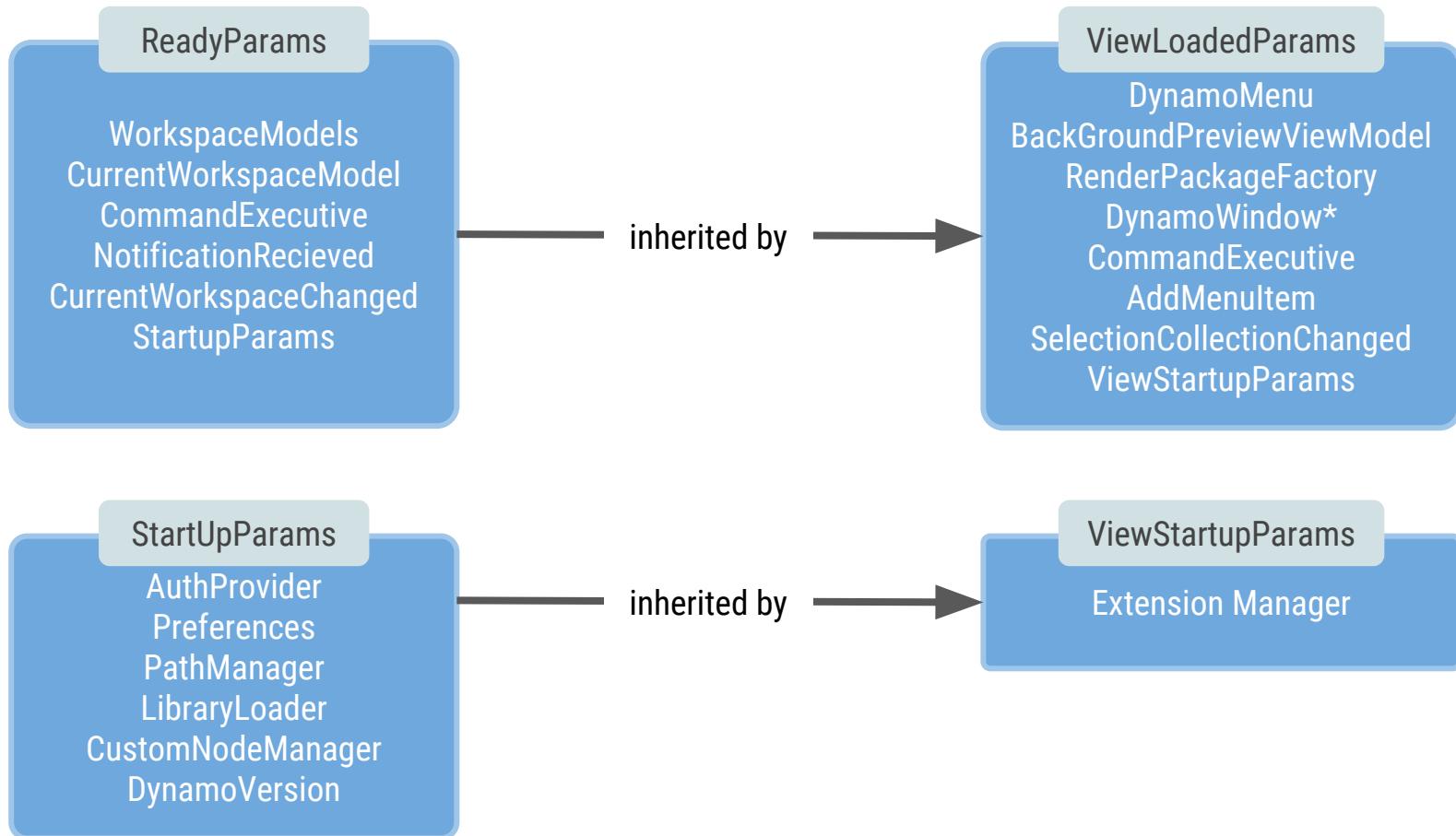


Loading Extensions from Packages



composition of extensions





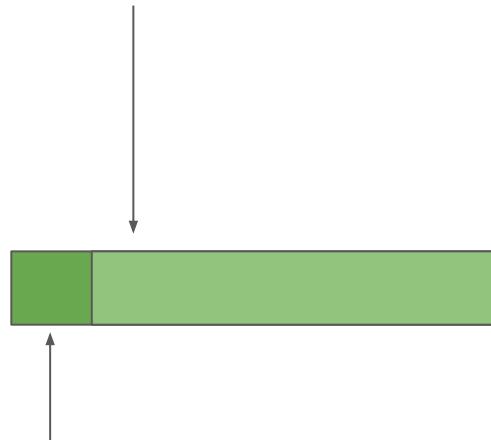
making stuff happen in Dynamo (Dynamo APIs)



Dynamo Model
Startup

extension lifecycle

Instantiate Extensions



Dynamo Model
Startup

extension lifecycle

Instantiate Extensions



Dynamo Model
Startup



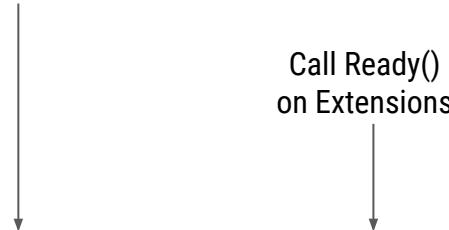
Call StartUp()
on
Extensions

StartUpParams

called when Dynamo starts loading

extension lifecycle

Instantiate Extensions



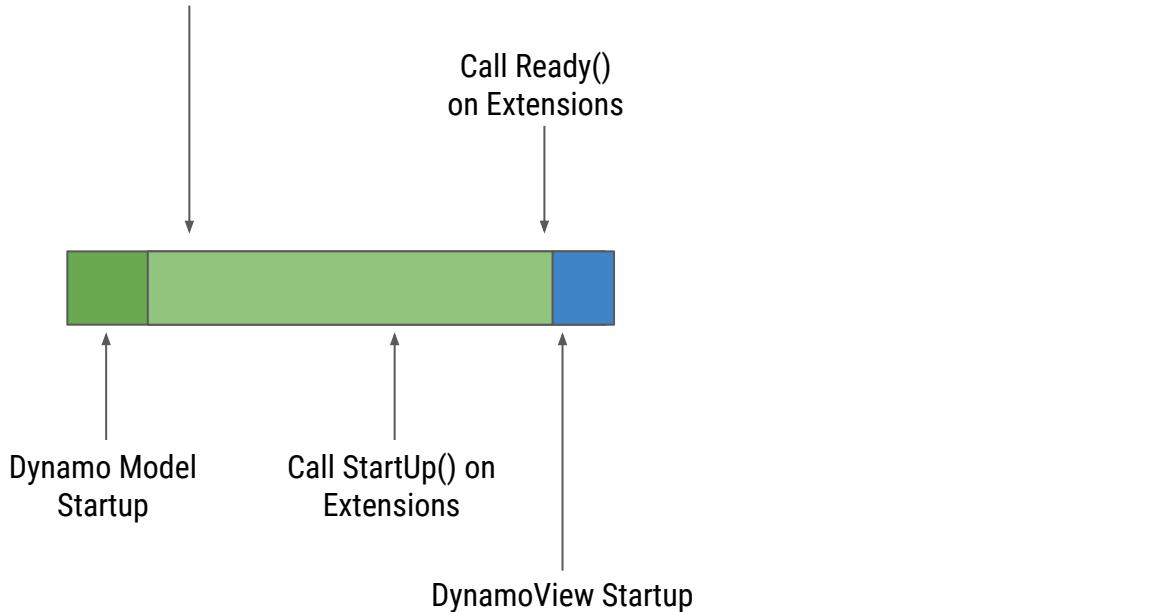
Dynamo Model
Startup

Call StartUp()
on Extensions

StartUpParams
called when Dynamo starts loading

extension lifecycle

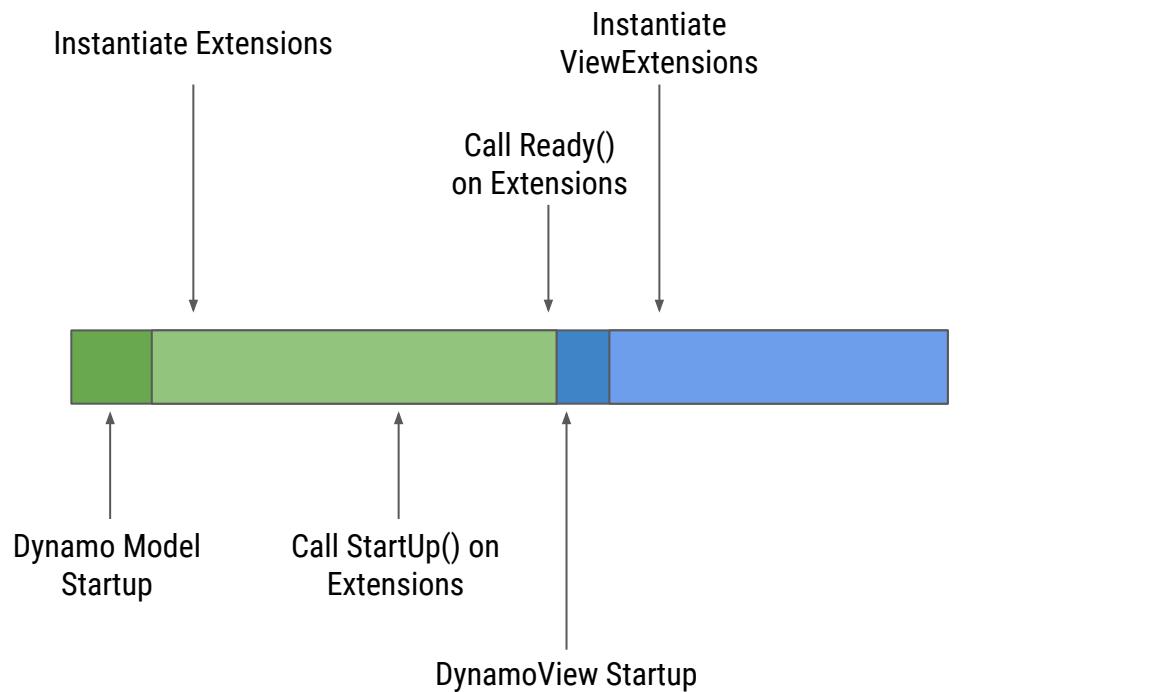
Instantiate Extensions



StartUpParams
called when Dynamo starts loading

ReadyParams & LoadedParams
called when Dynamo is finished loading

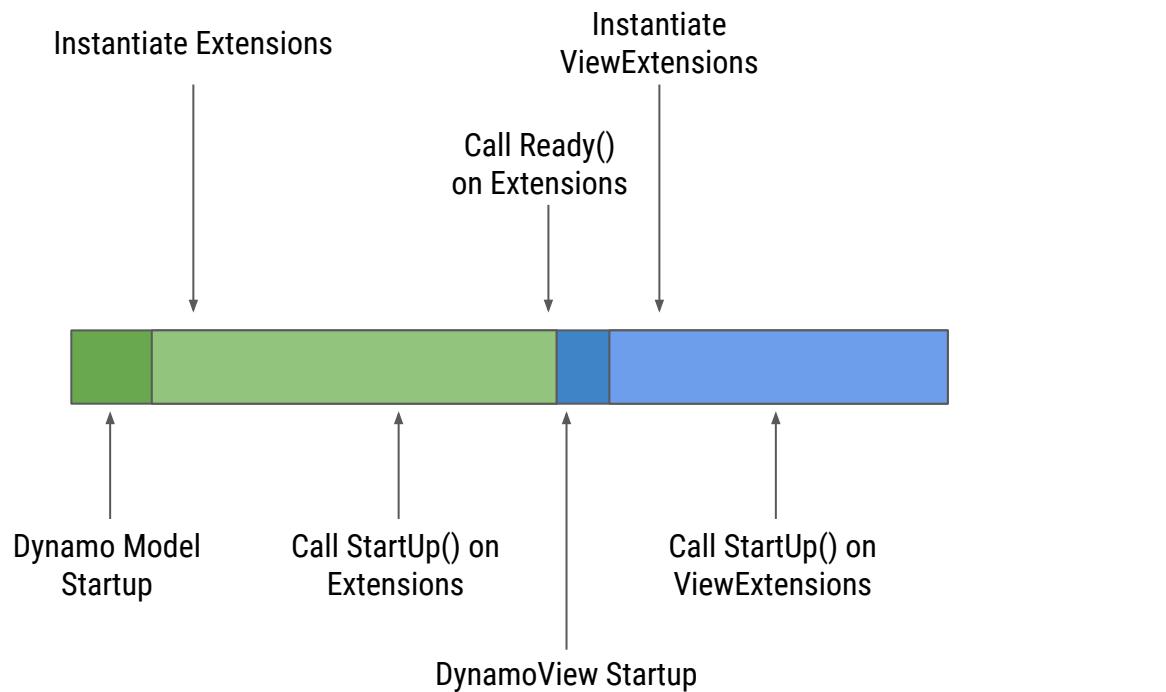
extension lifecycle



StartUpParams
called when Dynamo starts loading

ReadyParams & LoadedParams
called when Dynamo is finished loading

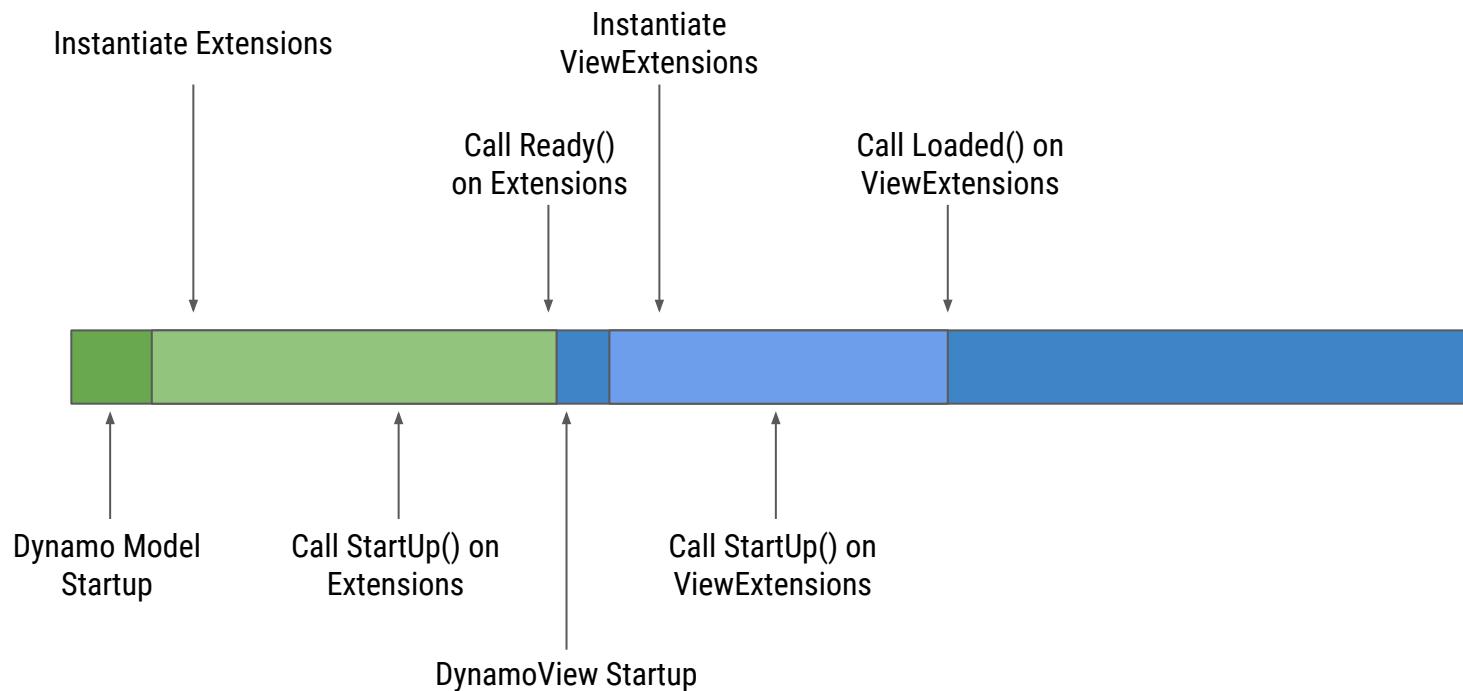
extension lifecycle



StartUpParams
called when Dynamo starts loading

ReadyParams & LoadedParams
called when Dynamo is finished loading

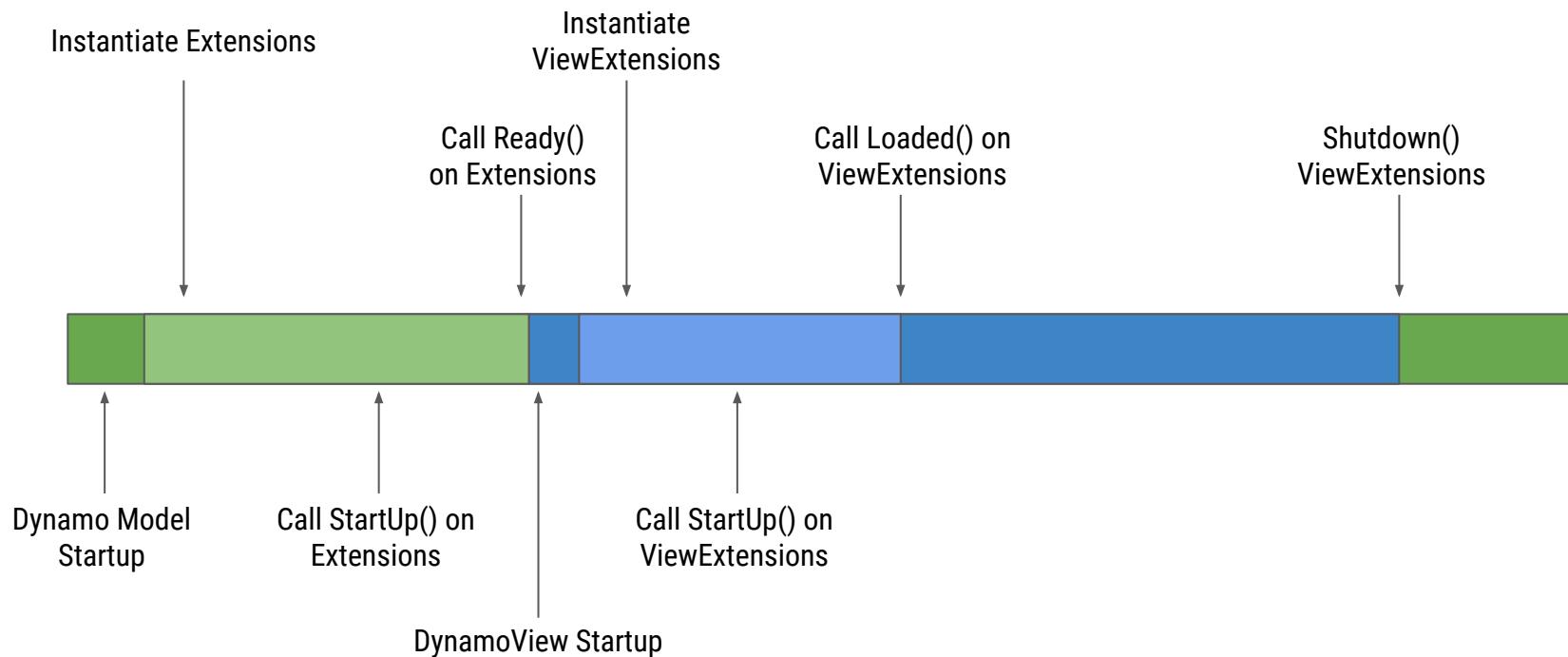
extension lifecycle



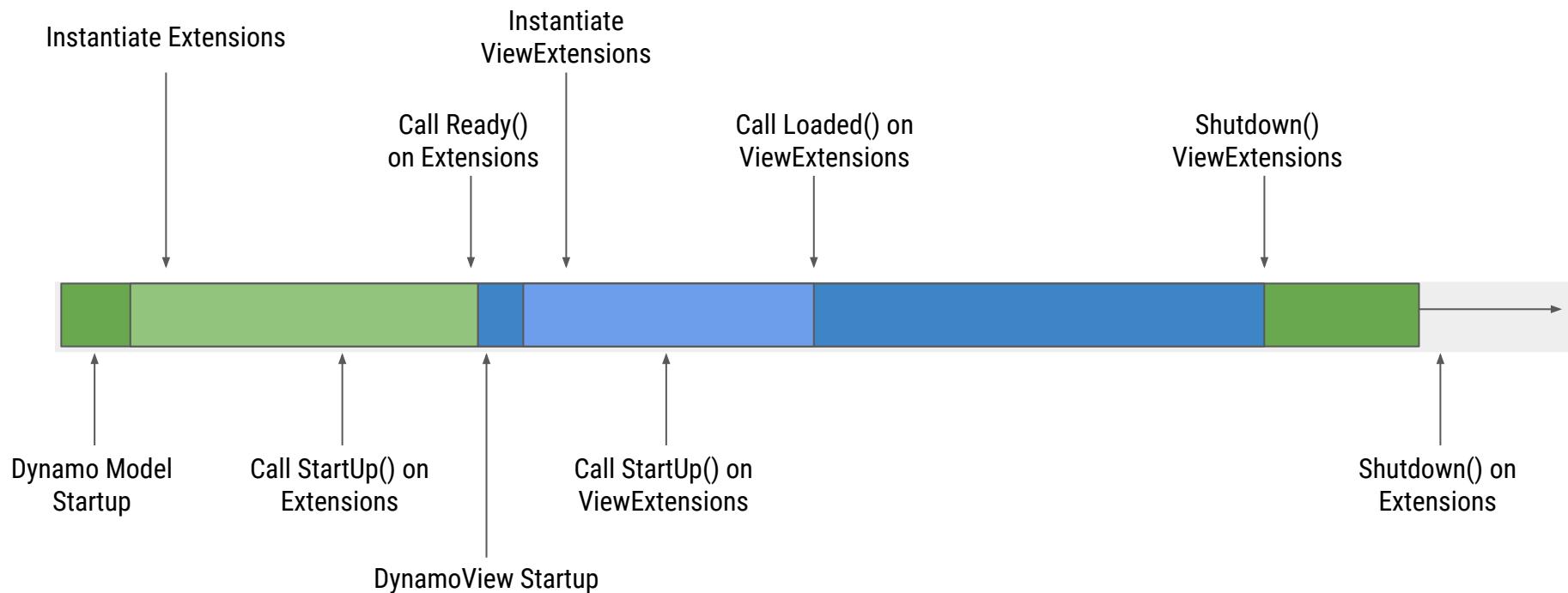
StartUpParams
called when Dynamo starts loading

ReadyParams & LoadedParams
called when Dynamo is finished loading

extension lifecycle



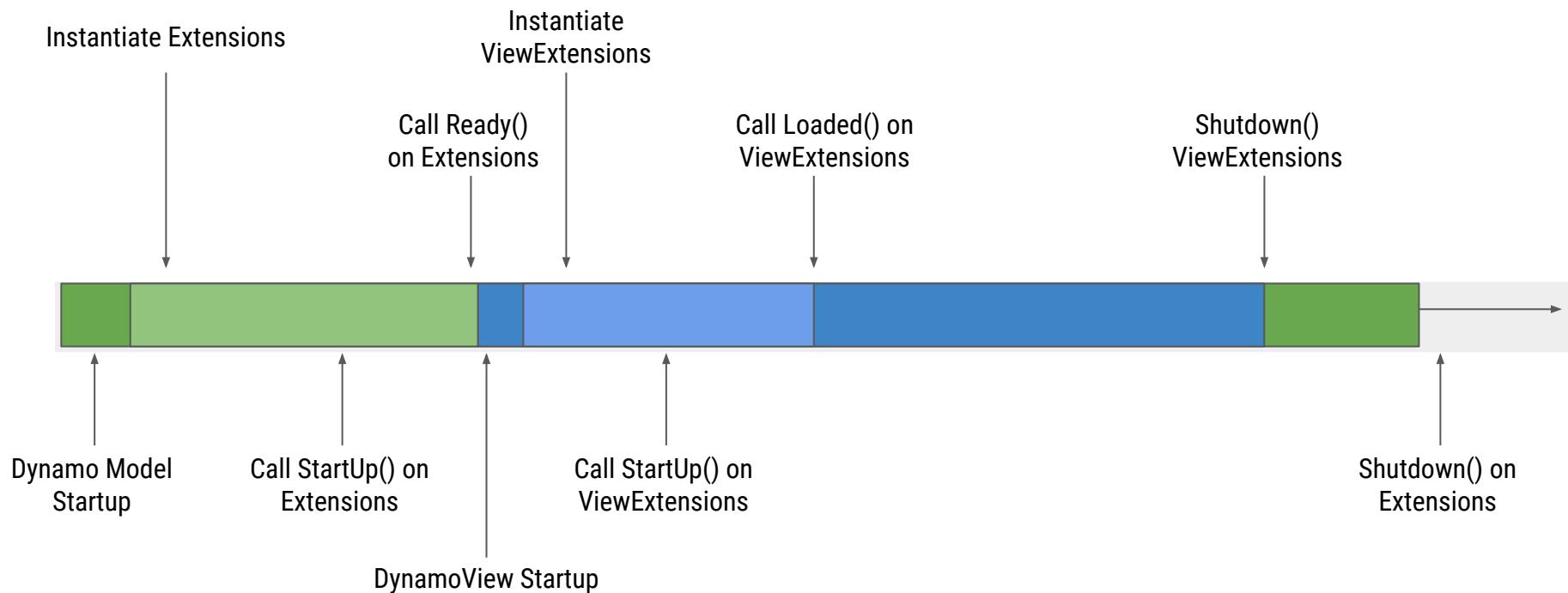
extension lifecycle



StartUpParams
called when Dynamo starts loading

ReadyParams & LoadedParams
called when Dynamo is finished loading

extension lifecycle



StartUpParams
called when Dynamo starts loading

ReadyParams & LoadedParams
called when Dynamo is finished loading

⚠️ Extensions loaded via **Packages**
will not have StartUp() called if they are
installed after DynamoStartup!!!!

extension lifecycle

WorkspaceModel

Nodes

Connectors

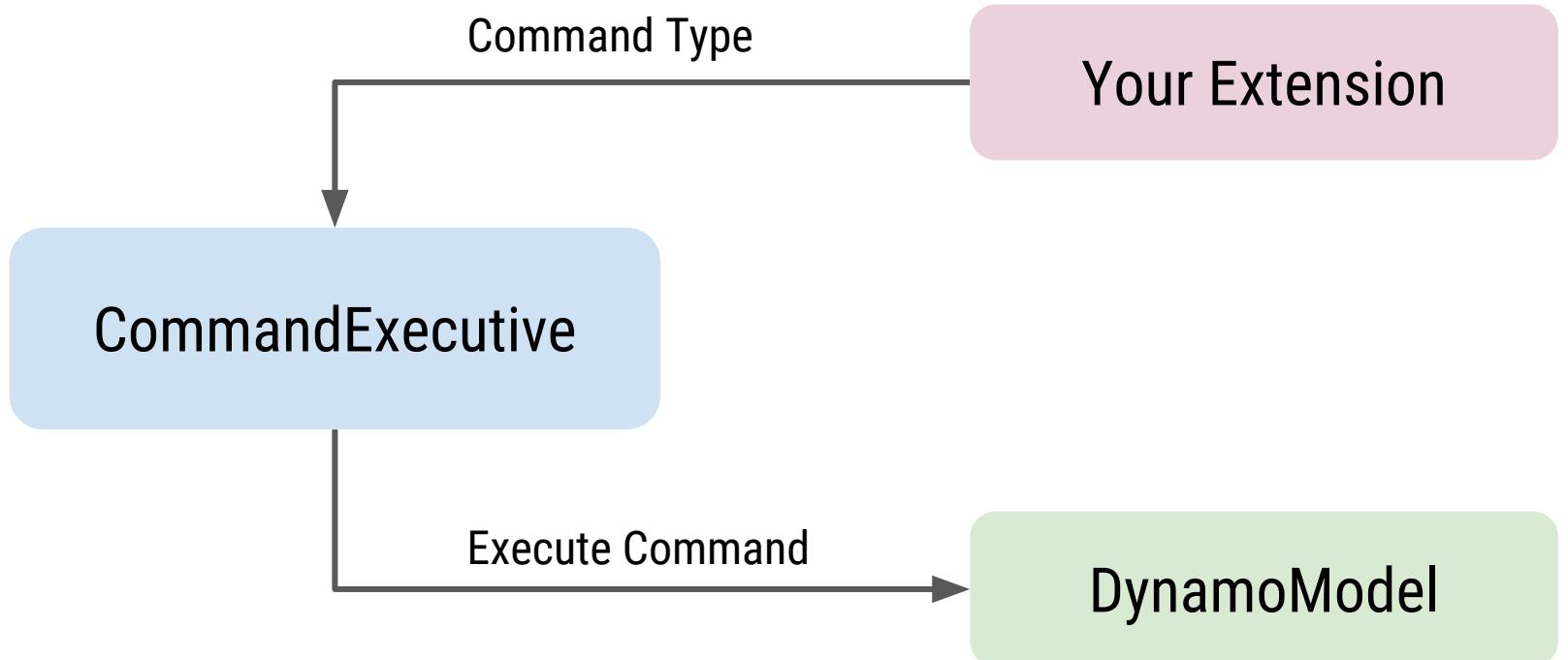
Annotations

Notes

Modification Events

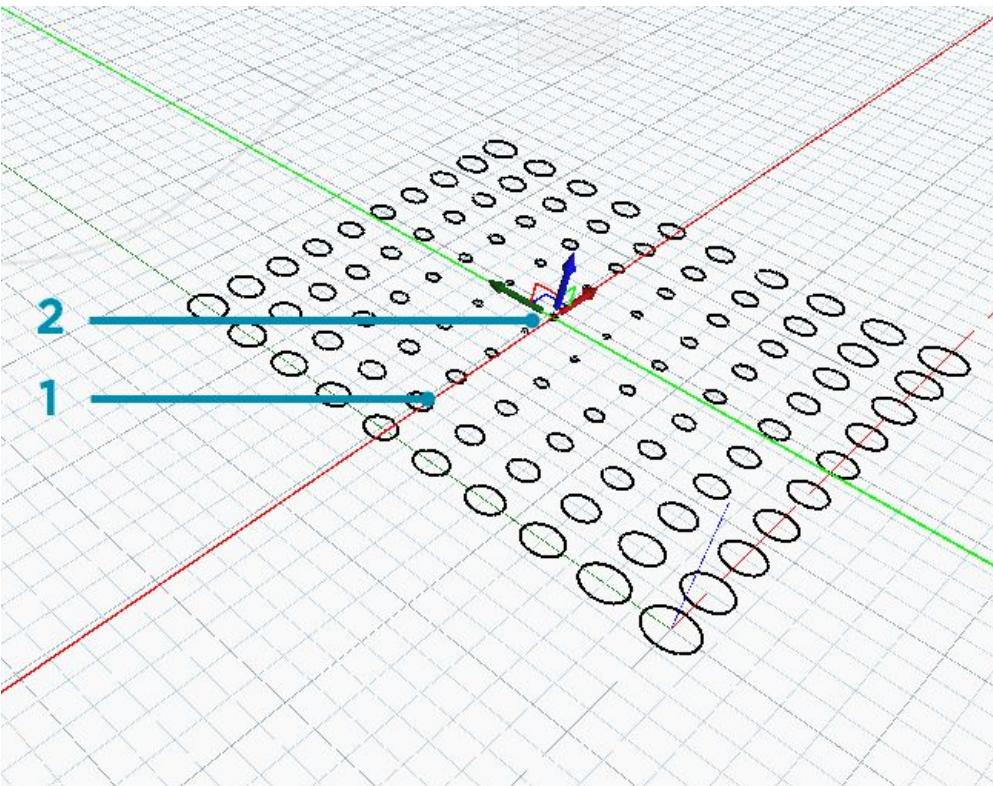
Cast to [HomeWorkspaceModel](#)
or
[CustomNodeWorkspaceModel](#)
for more APIs

CurrentWorkspaceModel



Recordable Command API for interacting with DynamoModel

BackgroundPreviewViewModel



IWatch3DViewModel

Events

ViewMouseMove

ViewCameraChanged

Properties

GetClickRay

GetCameraPosition

Methods

DeleteGeometryForNode

HighlightNodeGraphics

<https://www.fuget.org/packages/DynamoVisualProgramming.Core/>

.NET API 1,717,120 bytes

Search

Assemblies

- DesignScriptBuiltin.dll
- DSIronPython.dll
- DynamoApplications.dll
- DynamoCore.dll**
- DynamoInstallDetective.dll
- DynamoShapeManager.dll
- DynamoUtilities.dll
- ProtoCore.dll
- VMDataBridge.dll

Namespaces

- Dynamo
- Dynamo.Annotations
- Dynamo.Configuration
- Dynamo.Core
- Dynamo.Engine
- Dynamo.Engine.CodeCompletion
- Dynamo.Engine.CodeGeneration
- Dynamo.Engine.NodeToCode
- Dynamo.Exceptions
- Dynamo.Extensions**
- Dynamo.Graph
- Dynamo.Graph.Annotations
- Dynamo.Graph.Connectors
- Dynamo.Graph.Nodes
- Dynamo.Graph.Nodes.CustomNodes
- Dynamo.Graph.Nodes.NodeLoaders
- Dynamo.Graph.Nodes.ZeroTouch
- Dynamo.Graph.Notes
- Dynamo.Graph.Presets
- Dynamo.Graph.Workspaces
- Dynamo.Interfaces

★ IExtension

public interface **IExtension** : IDisposable

An extension to the model layer of Dynamo

Documentation Code

`string Name { get; }`

A name for the Extension. This is used for more user-readable logging.

`string UniqueId { get; }`

A unique id for this extension instance. There may be multiple instances of the same type, but the application will *not* allow two instances to coexist with the same id.

`void Ready(ReadyParams sp)`

Action to be invoked when the Dynamo has started up and is ready for user interaction. This action is guaranteed to be called even if the extension is installed after startup. Exceptions thrown from this method will be caught by Dynamo and logged.

`void Shutdown()`

Action to be invoked when shutdown has begun.

`void Startup(StartupParams sp)`

Action to be invoked when Dynamo begins to start up. This action is *not* guaranteed to be invoked unless the extension is already installed at startup. Exceptions thrown from this method will be caught by Dynamo and logged.

Dirty Tricks

making more stuff happen

`ViewLoadedParams.DynamoWindow.DataContext as DynamoViewModel`



⚠ Don't do this unless you have to - it may stop working!

let's get started.

For this demo, we are going to run our extension in the installed **Dynamo Sandbox**.

But, you can always build it yourself!

DynamoDS / Dynamo

Code Issues 611 Pull requests 10 Projects 2 Wiki Insights

Open Source Graphical Programming for Design <http://dynamobim.org>

28,220 commits 70 branches 0 releases 69 contributors

Branch: master New pull request Create new file Upload files Find file Clone

ramramps committed 8 hours ago Merge pull request #8502 from ramramps/LibrariesJS-update ...

File	Description
.github	Fixing a typo about DynamoRevit repo
doc	Updated the year information (#7673)
extern	Support LibG for ASM224 - Cherry-pick from #8471 (#8471)
src	Merge pull request #8502 from ramramps/LibrariesJS-update
test	Merge pull request #8494 from mjkirschner/keepConnectorIds
tools	add Builtin.dll to Core Nuget package (#8426)
.gitattributes	One time renormalization of line endings.
.gitignore	Update .gitignore
.gitmodules	checking ssh key
.travis.yml	Update travis
CONTRIBUTING.md	Fixing typo about DynamoRevit
LICENSE.txt	Clarify dependency licenses and their provenance
README.md	Updated pull request link to new wiki page
appveyor.yml	resolved merge conflicts
dynamo-nuget.config	Set nuget dependency version to highest.
dynamo_sublime	Add a sublime text project.

Clone with HTTPS <https://github.com/DynamoDS/DynamoDS.git>

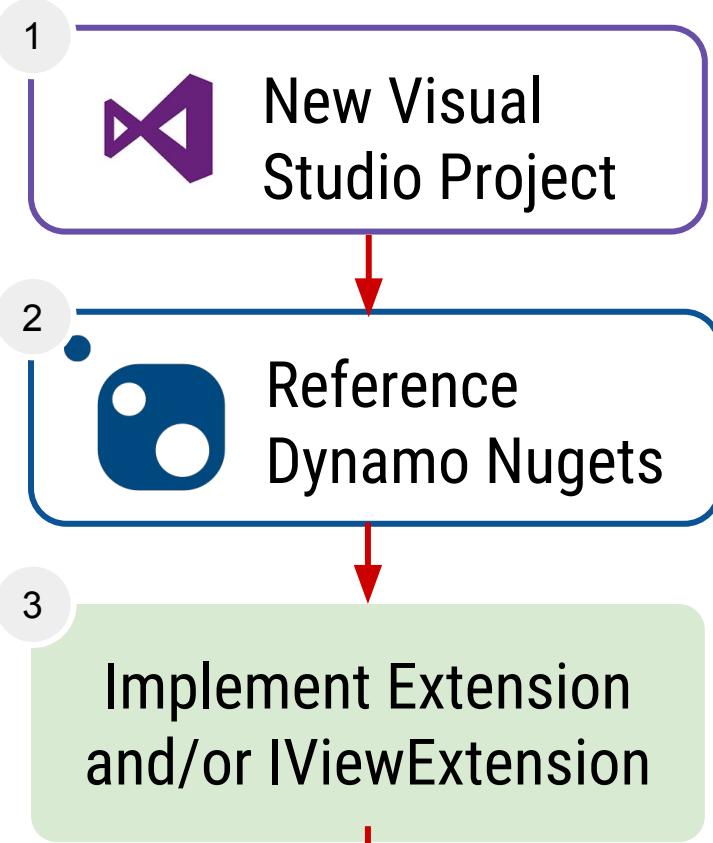
Open in Desktop Download

<https://github.com/DynamoDS/DeveloperWorkshop>

First Extension

Events

MenuItems



Example extension creation workflow

4

Respond to some
Dynamo events



5

Add Menu Item



6

Load extension in
Dynamo

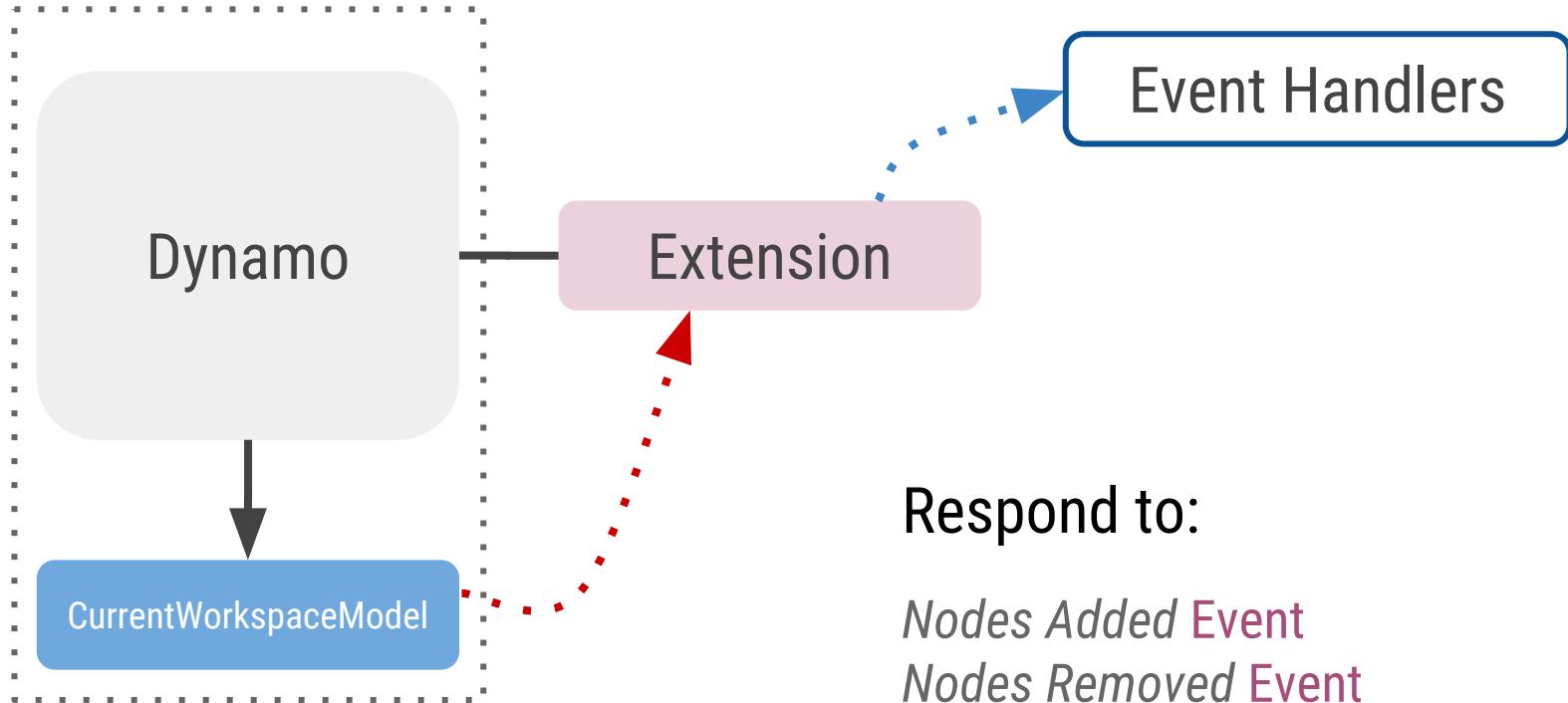
Example extension creation workflow

Hello Dynamo.csproj

Events.cs
ExtensionExample.cs
ViewExtensionExample.cs

pkg.json
package.config

Extension Manifests
(*.xml)



Respond to:

Nodes Added Event

Nodes Removed Event

Current Workspace Changed Event

ExtensionExample: IExtension

Startup(StartupParams)

Ready(ReadyParams)

Shutdown()

1

2

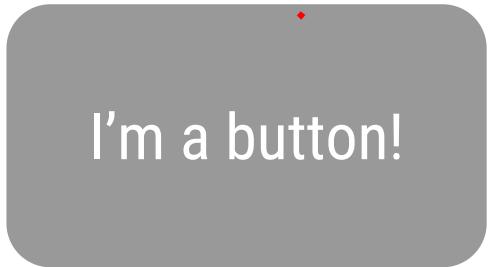
MessageBox.Show("ready")

RegisterEventHandlers()

1

UnRegisterEventHandlers()

Click Event



events

class event subscribe handler
function

Button.Click += HandleClick;

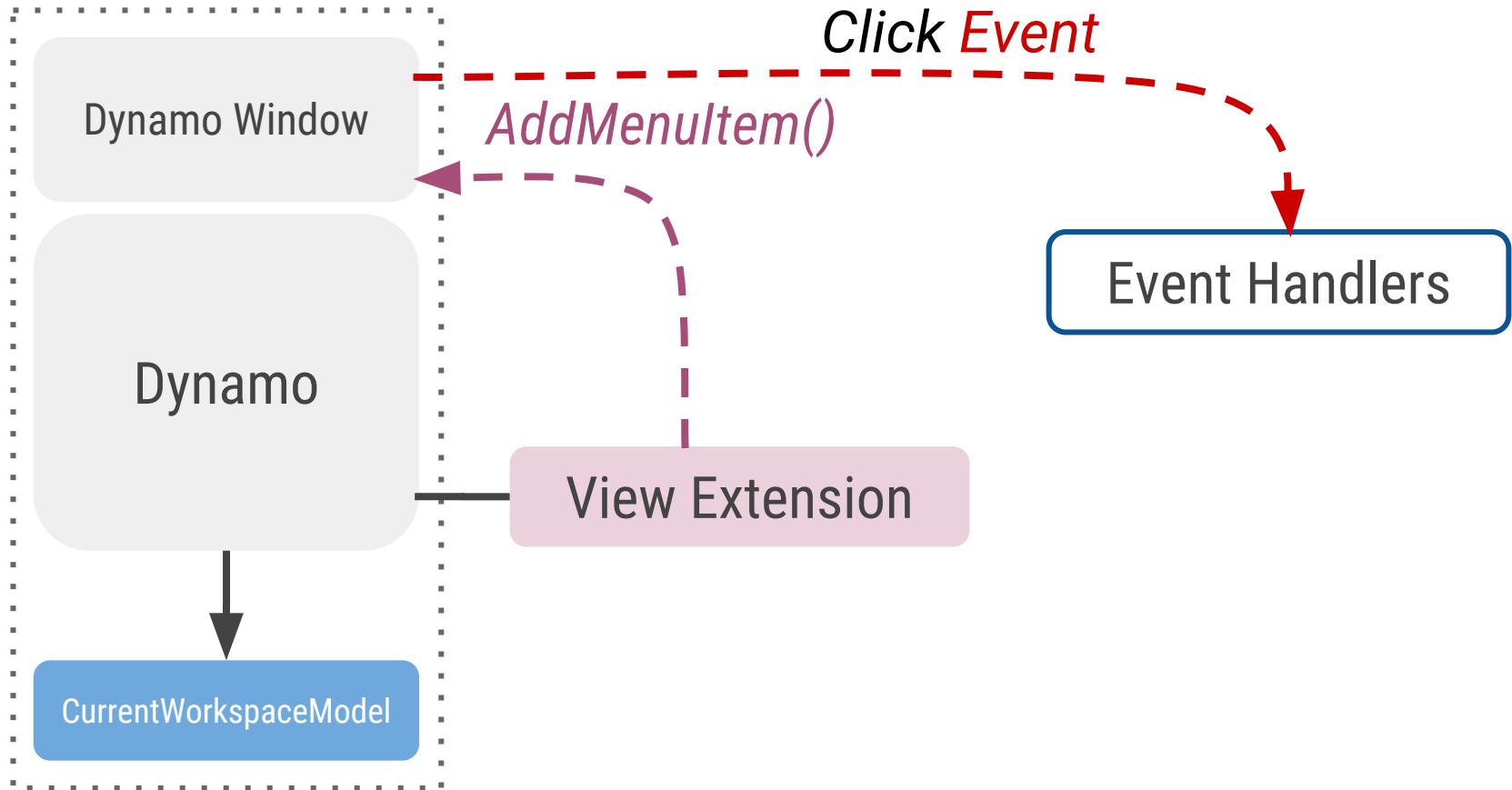
Button.Click -= HandleClick;

events

unsubscribe

```
public static void RegisterEventHandlers(ReadyParams dynamoReadyParams)
{
    dynamoReadyParams.CurrentWorkspaceChanged += OnCurrentWorkspaceChanged;
    dynamoReadyParams.CurrentWorkspaceModel.NodeAdded += OnNodeAdded;
    dynamoReadyParams.CurrentWorkspaceModel.NodeRemoved += OnNodeRemoved;
}
```

events



Hello Dynamo extension diagram

ViewExtensionExample: IViewExtension

Startup(StartupParams)

Loaded(ViewLoadedParams)

Shutdown()

1

2

MessageBox.Show("ready")

MakeMenuItems()



File Edit View Packages Settings Help Debug



AU Workshop

Say Hello

Menu Items

```
public void MakeMenuItems()
{
    extensionMenu = new MenuItem { Header = "AU Workshop" };

    //and now we add a new sub-menu item that says hello when clicked
    var sayHelloMenuItem = new MenuItem { Header = "Say Hello" };

    // finally, we need to add our menu to Dynamo
    viewLoadedParams.dynamoMenu.Items.Add(extensionMenu);

    ...
}
```



File Edit View Packages Settings Help Debug

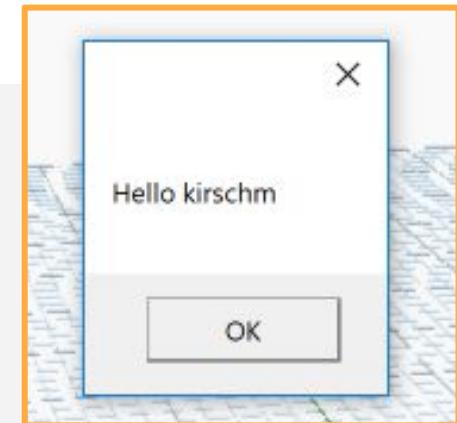


AU Workshop

Say Hello

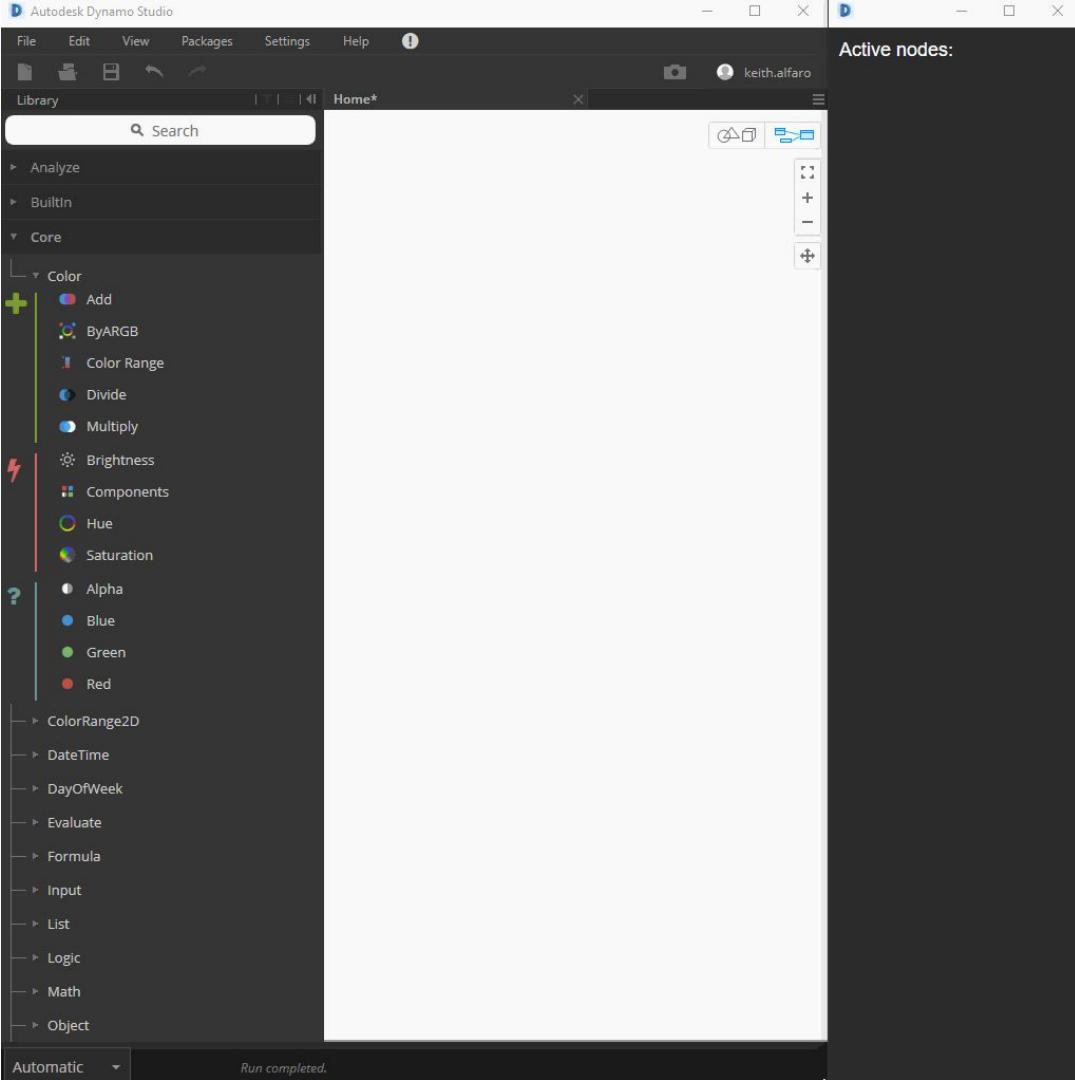
sayHelloMenuItem

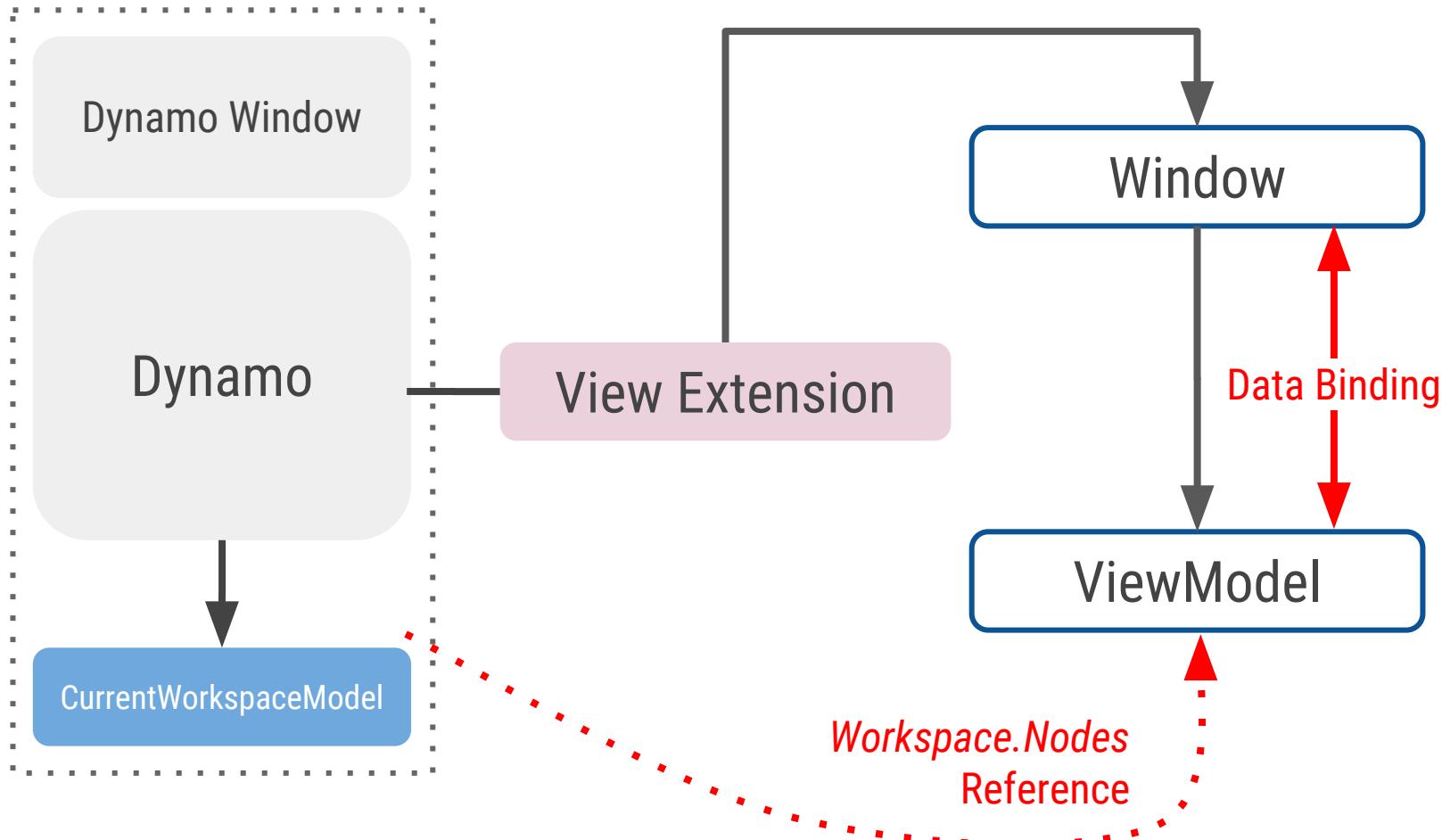
```
public void MakeMenuItems()
{
...
    sayHelloMenuItem.Click += (sender,args) =>
    {
        MessageBox.Show("Hello" + Environment.UserName);
    };
}
```



For more samples, check out
the DynamoSamples repo:

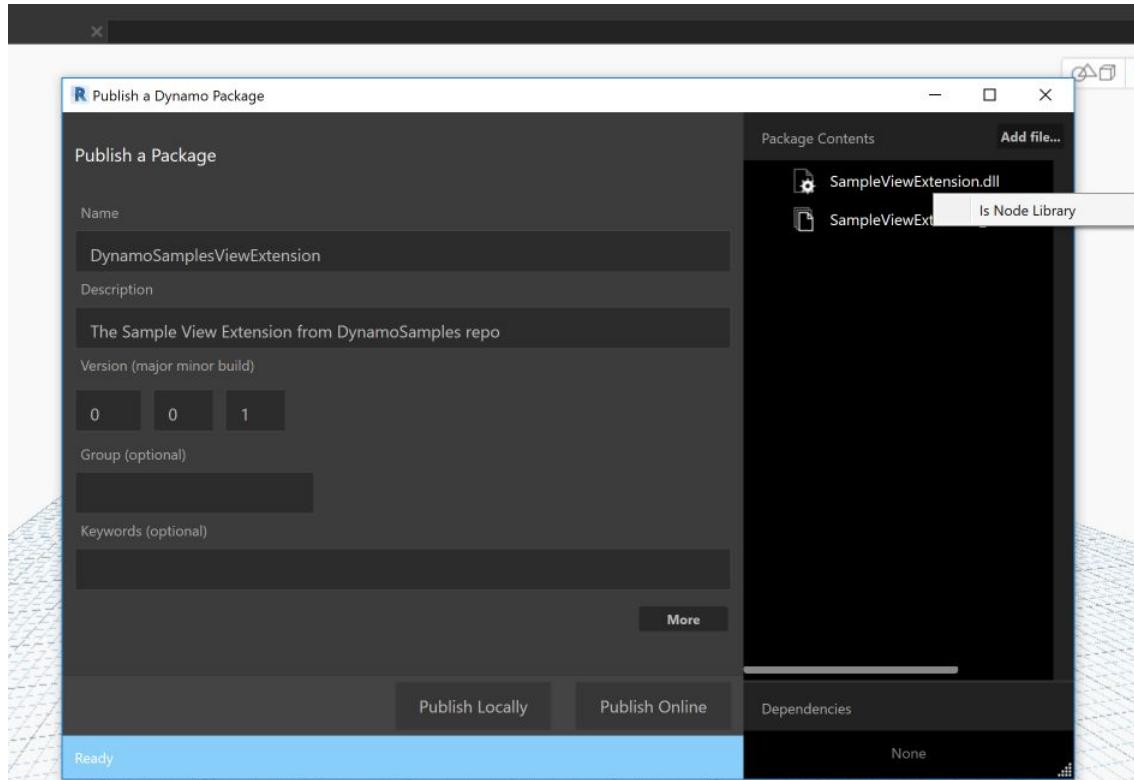
[https://github.com/
DynamoDS/
DynamoSamples](https://github.com/DynamoDS/DynamoSamples)





Dynamo extension diagram

deploy an extension *using a Dynamo package*



dynamo commands

ModelBasedRecordableCommand: RecordableCommand

ModelGuids

Command: RecordableCommand

Serialize()
Deserialize()
Execute()

Other Commands

OpenFileCommand
RunCancelCommand
CreateNodeCommand
SelectModelCommand
CreateNoteCommand

Dynamo\src\DynamicCore\Models\RecordableCommands.cs

Execute model commands

```
p.CommandExecutive.ExecuteCommand  
(new DynamoModel.ForceRunCancelCommand(true, false),  
this.UniqueId,  
this.Name)
```

Executing Commands on the Dynamo Model.

```
var VM = ViewLoadedParams.DynamoWindow.DataContext as DynamoViewModel;  
//workspace view model command  
VM.CurrentSpaceViewModel.ResetFitViewToggleCommand.Execute(null);  
  
//dynamoViewModel command  
VM.AddToSelectionCommand.Execute(someNode);
```

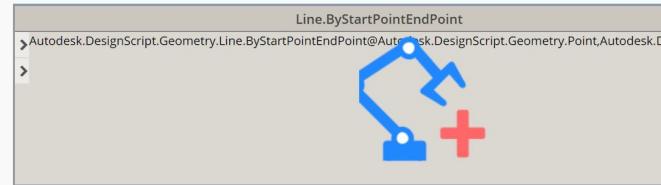
⚠ Execute viewModel commands ... not guaranteed!

Executing Commands on the Dynamo View Model.

injecting UI into other views

Object.IsNull

obj	> DSCore.Object.IsNull@var	bool
AUTO		



GeometryColor.ByGeometryColor

geometry	> Modifiers.GeometryColor.ByGeometryColor@Autodesk.DesignScript.Geometry.Geometry,DSCore.Color	GeometryColor
color	>	AUTO

Mesh.FaceIndices

mesh	> Autodesk.DesignScript.Geometry.Mesh.FaceIndices	IndexGroup[]
AUTO		



<https://github.com/mjkkirschner/DynamoSamples/tree/extensionWorkshop>

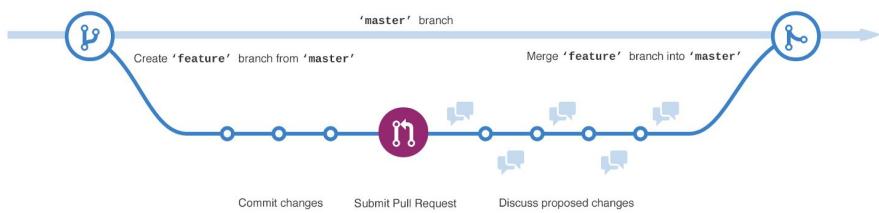
```
1 reference | 0 changes | 0 authors, 0 changes
public void Loaded(ViewLoadedParams p)
{
    this.loadedParams = p;
    p.DynamoWindow.LayoutUpdated += DynamoWindow_ContentRendered;

}

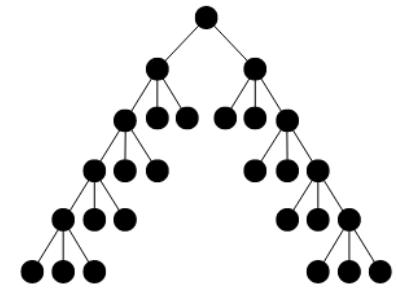
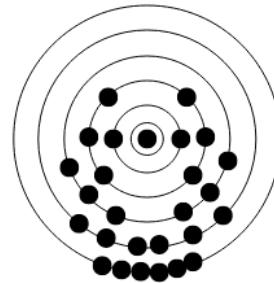
1 reference | 0 changes | 0 authors, 0 changes
private void DynamoWindow_ContentRendered(object sender, EventArgs e)
{
    var nodeViews = this.loadedParams.DynamoWindow.FindVisualChildren<NodeView>();
    foreach(var nv in nodeViews)
    {
        //if there is no existing label, add it.
        if (nv.inputGrid.Children.OfType<TextBlock>().Where(x => x.Name == "typeLabel").Count() == 0)
        {
            var nodeTypeLabel = new TextBlock();
            nodeTypeLabel.Name = "typeLabel";
            nodeTypeLabel.Text = nv.ViewModel.NodeModel.CreationName;
            nv.inputGrid.Children.Add(nodeTypeLabel);
        }
    }
}
```

⚠ A *hack* to illustrate injecting some wpf control whenever the view is updated...
performance implications and other caching issues with this code.

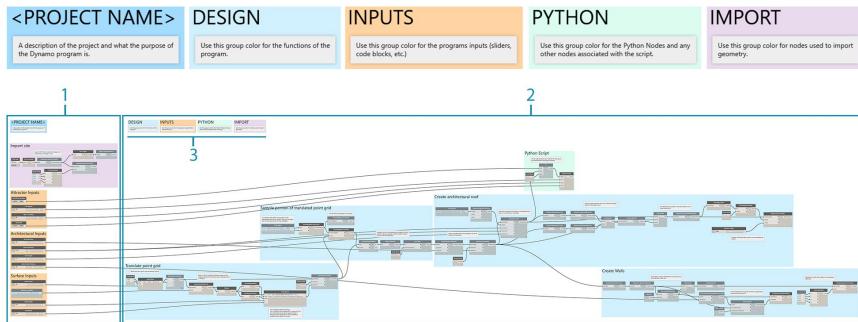
Tracking graph changes



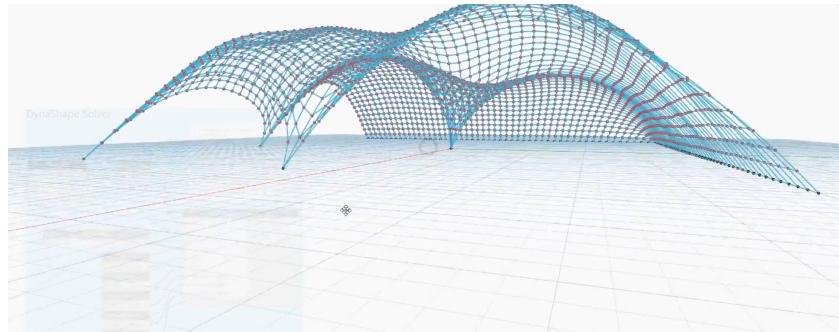
Alternative UIs



Templates



Geometry Interaction



Developer Challenges

Testing, Testing, Testing!!!!

dynamo source

DynamoDS / Dynamo

Watch ▾ 175 Star 638 Fork 368

Code Issues 548 Pull requests 13 Projects 2 Wiki Insights Settings

Open Source Graphical Programming for Design <http://dynamobim.org> Edit

Manage topics

28,667 commits 78 branches 0 releases 70 contributors View license

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

mjkirschner Merge pull request #9152 from mjkirschner/master_libGLoadingOldClient ... Latest commit f12158a 2 days ago

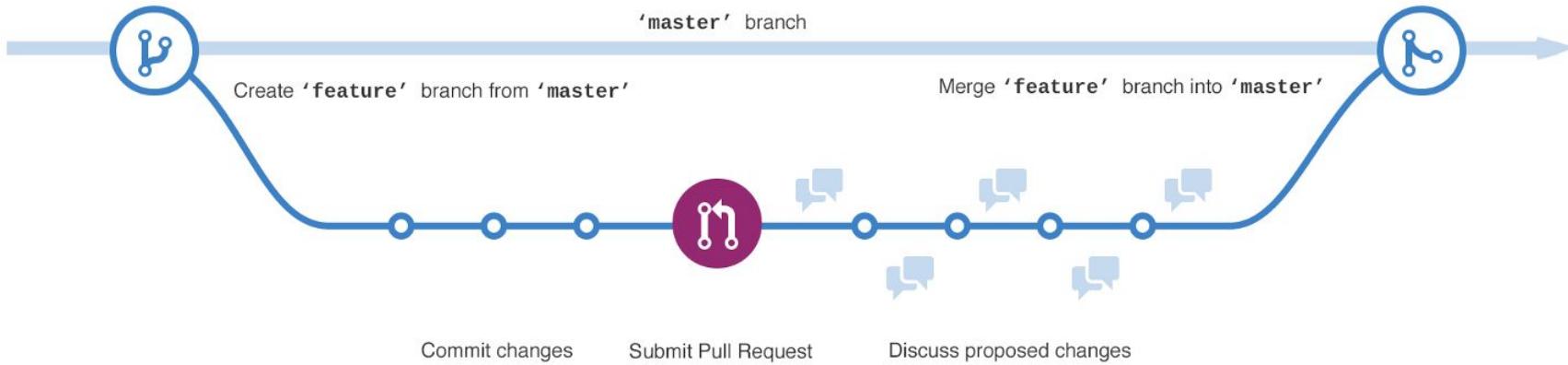
.github Fixing a typo about DynamoRevit repo a year ago

doc update revit sample files 6 months ago

extern Remove extraneous gettext binaries from LibG223 folder (#9155) 2 days ago

src review comments and tests 2 days ago

code review



no regressions!

Commits on Feb 13, 2018

[QNTM-3158: Display multi-return node previews in order of output ports \(#8548\)](#) ...



aparajit-pratap committed 12 hours ago

Commits on Feb 12, 2018

[QNTM-2116: Check for persistent warning as well as plain warning \(#8549\)](#) ...



ColinDayOrg committed a day ago

[QNTM-3340: Fix crash with CBN in error state connected to downstream ... \(#8550\)](#) ...



aparajit-pratap committed 2 days ago

[update LibG built against ASM224 for Revit2019 \(#8558\)](#) ...



aparajit-pratap committed 2 days ago

[DYN-899: Make symbols local to imperative block scope \(#8306\)](#) ...



aparajit-pratap committed 2 days ago

[Merge pull request #8554 from pboyer/fixoverlap](#) ...



pboyer committed 2 days ago

Process for DesignScript Language Changes

Following is a process we will follow in making language changes. The outcome will be better considered changes, user facing documentation of the impact of these changes, and procedures for users to follow in upgrading.

1. Change is proposed
2. Motivation
3. Discussion of how the change aligns or diverges from the Goals and Principles of the language
4. Assessment of change
 - What is change: Technical description of the change
 - Assess who is affected: check instrumentation to see breadth of usage and kind of usage if possible (mission critical or cosmetic)
 - Long term user experience/user interface impact
 - Performance assessment
 - Evaluate against benchmark datasets to assess breakages
 - Migration strategy and/or method to introduce error messaging
 - Implications for the visual expression of the language
 - Does the change align with strategies for distributed/cloud based compute?
 - Effort needed to make the change itself
 - Risk (what other language features might break) and Benefit
5. Go/No Go decision
 - based on 1 to 2 page doc of #1-4.
 - Decided in 1 meeting with stakeholders: User Experience Design, Software Development, and Product Management representatives
6. Release
 - Large changes require soft release and pre-release announcements for people to update their code
 - Monitor the implementation of changes ahead of release

test, test, test

like clash detection or model checking

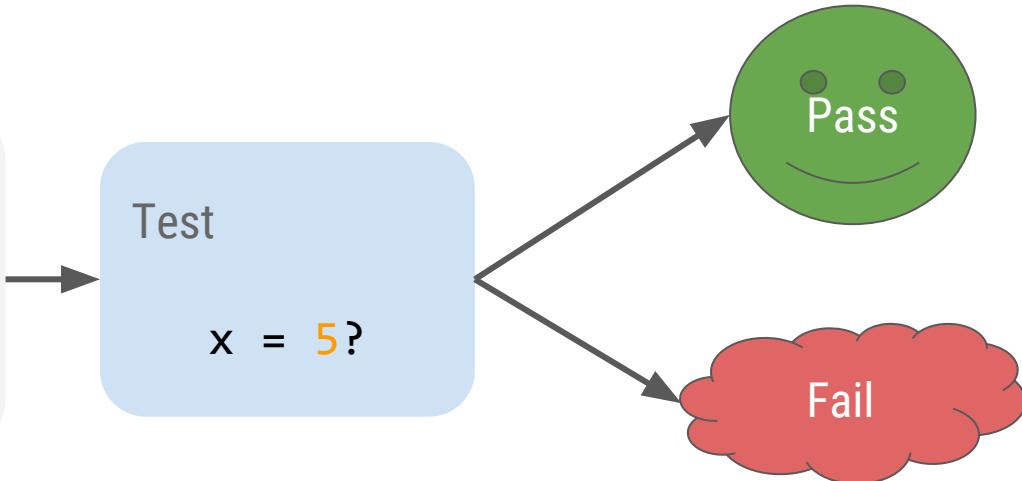
Bug Testing
command files and NUnit

Recorded Dynamo Sessions
Zeenyx, Selenium, etc

Unit Testing
methods, small interactions

some tiny bit of code!!!

```
x = Line.Length();
```





developer.dynamobim.org

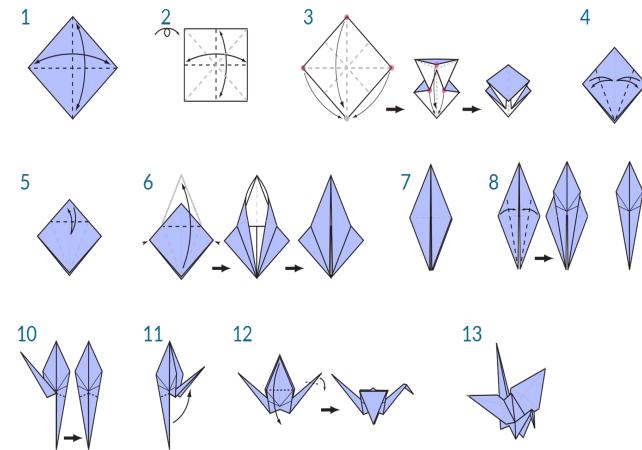
dynamobim.org

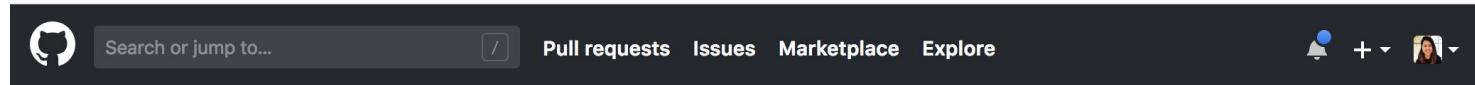
dictionary.dynamobim.org

primer.dynamobim.org

forum.dynamobim.com

@DynamoBIM





Code Issues 489

Pull requests 15

Projects 2

Wiki

Insights

Settings

Filters

is:pr is:open

Labels

Milestones

New pull request

15 Open 6,445 Closed

Author

Labels

Projects

Milestones

Reviews

Assignee

Sort

LibG Binaries

Help Us, Help You!

Prevent crash when starting DynamoSandbox without ASM

#9210 opened 12 days ago by Dewb • Approved 1 of 7

2

KeepListStructure list@level functionality should work.

#9182 opened 23 days ago by mjkkirschner 0 of 8

Update Image.Pixels to read full range of image.

#9181 opened 23 days ago by mjkkirschner 0 of 7

QNTM-5415 Expose ToJson() and OpenJsonFileFromPath() in DynamoCore as Public methods

#9161 opened 29 days ago by saintentropy 7 of 10

6

5909 pre-fill Github issue reports with crash details

#9150 opened on Oct 5 by radumg 6 of 7

18

Render precision save

#9108 opened on Sep 19 by DmytroMuravskyi 2 of 7

2



Questions?

BROADWAY
HOLLYWOOD

CAPITOL

Break Time!!!

Breakout Sessions

10:30 AM

View Extensions for C# Beginners Murano 3302, Level 3

Build Custom User Interfaces Murano 3303, Level 3

Control Dynamo from the Web Murano 3304, Level 3

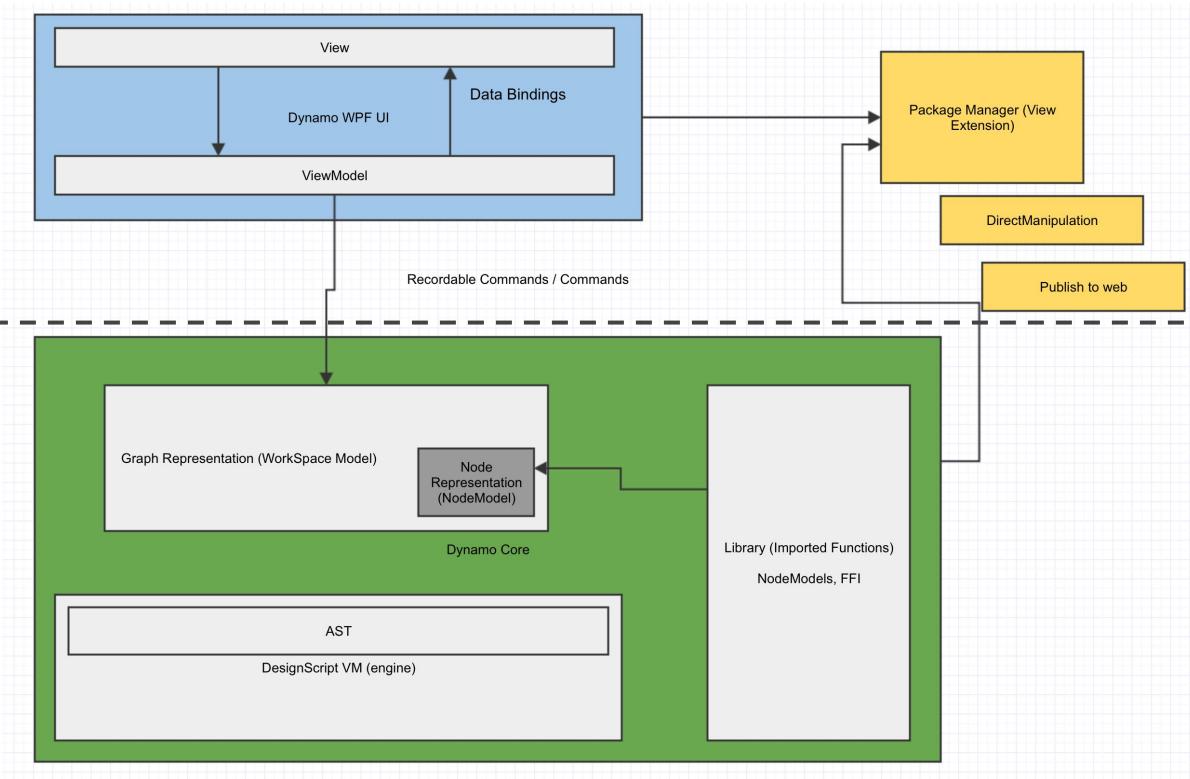
Leveraging Speckle in Dynamo San Polo 3404, Level 3

Custom Graphics Visualization San Polo 3405, Level 3

Dynamo Architecture(*a brief aside*)

DynamoCoreWPF.dll

DynamoCore.dll



Extensions Architecture

Extension LifeCycle:

StartUp(Params): called when Dynamo starts loading.

Ready/Loaded(Params): called when Dynamo is finished loading.

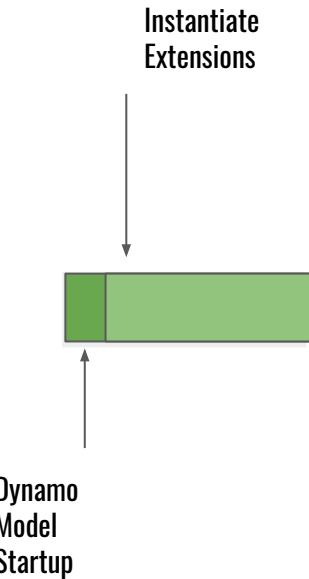


Extensions Architecture

Extension LifeCycle:

StartUp(Params): called when Dynamo starts loading.

Ready/Loaded(Params): called when Dynamo is finished loading.

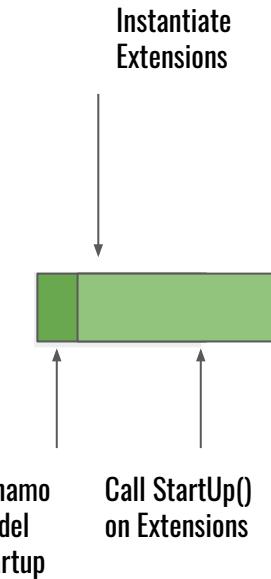


Extensions Architecture

Extension LifeCycle:

StartUp(Params): called when Dynamo starts loading.

Ready/Loaded(Params): called when Dynamo is finished loading.

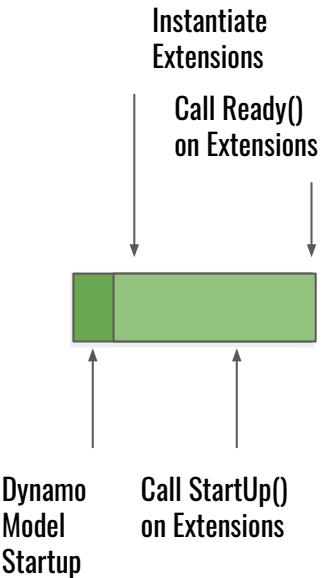


Extensions Architecture

Extension LifeCycle:

StartUp(Params): called when Dynamo starts loading.

Ready/Loaded(Params): called when Dynamo is finished loading.

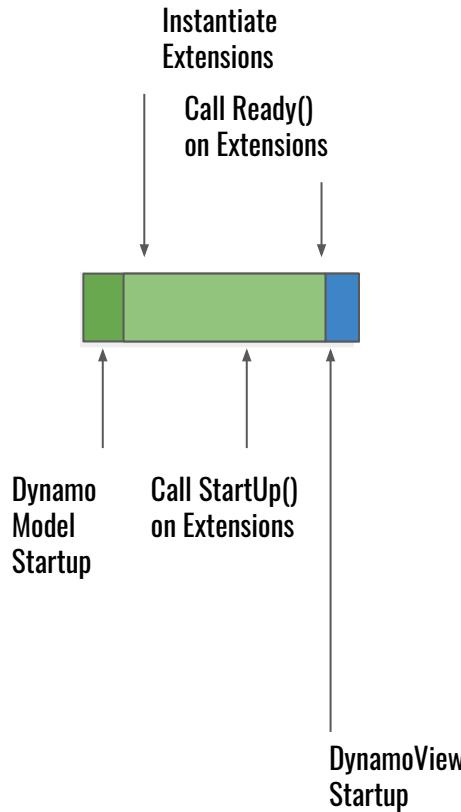


Extensions Architecture

Extension LifeCycle:

StartUp(Params): called when Dynamo starts loading.

Ready/Loaded(Params): called when Dynamo is finished loading.

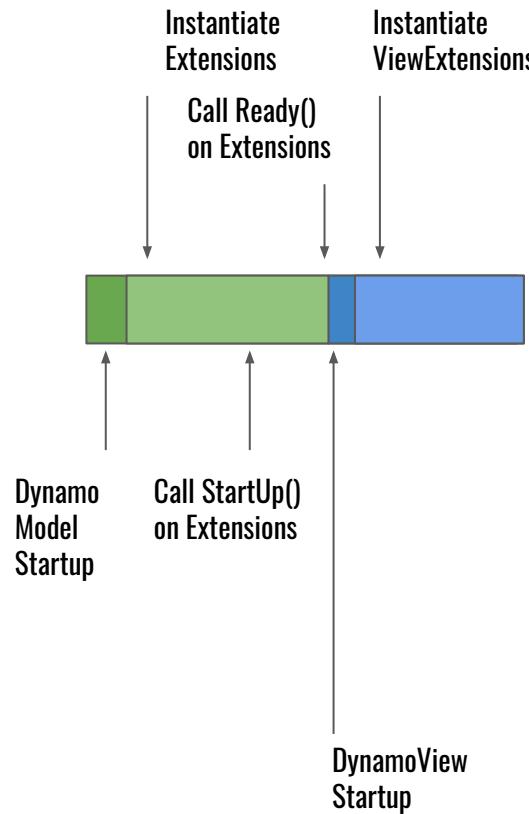


Extensions Architecture

Extension LifeCycle:

StartUp(Params): called when Dynamo starts loading.

Ready/Loaded(Params): called when Dynamo is finished loading.

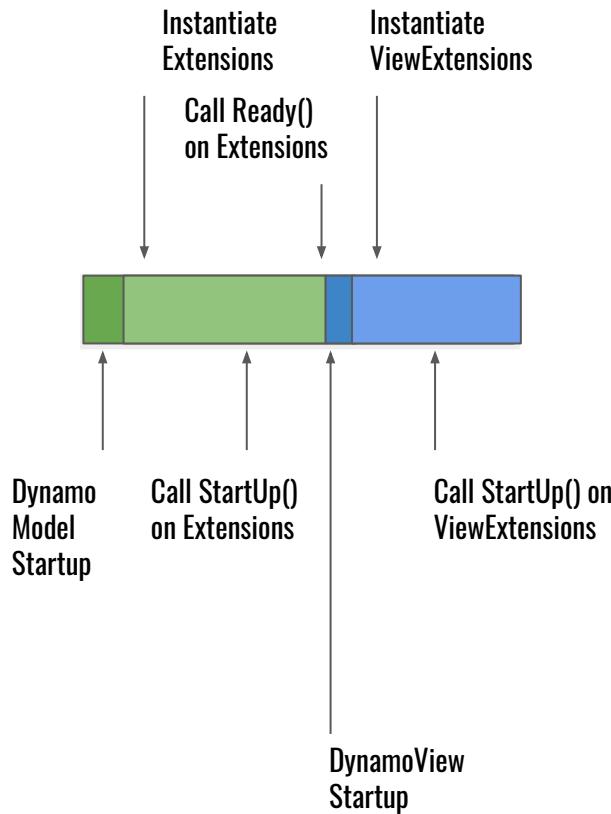


Extensions Architecture

Extension LifeCycle:

StartUp(Params): called when Dynamo starts loading.

Ready/Loaded(Params): called when Dynamo is finished loading.

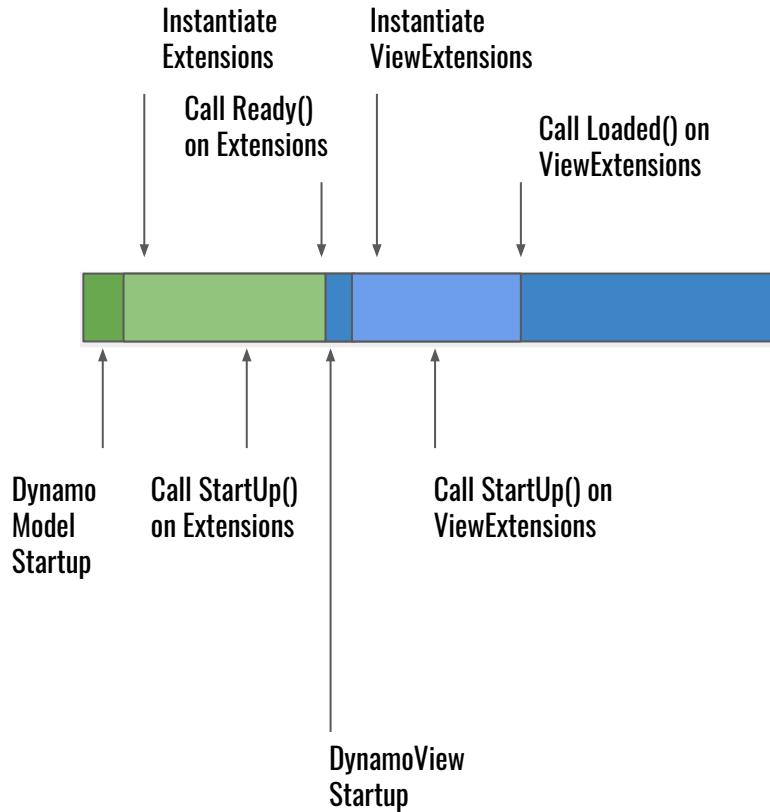


Extensions Architecture

Extension LifeCycle:

StartUp(Params): called when Dynamo starts loading.

Ready/Loaded(Params): called when Dynamo is finished loading.

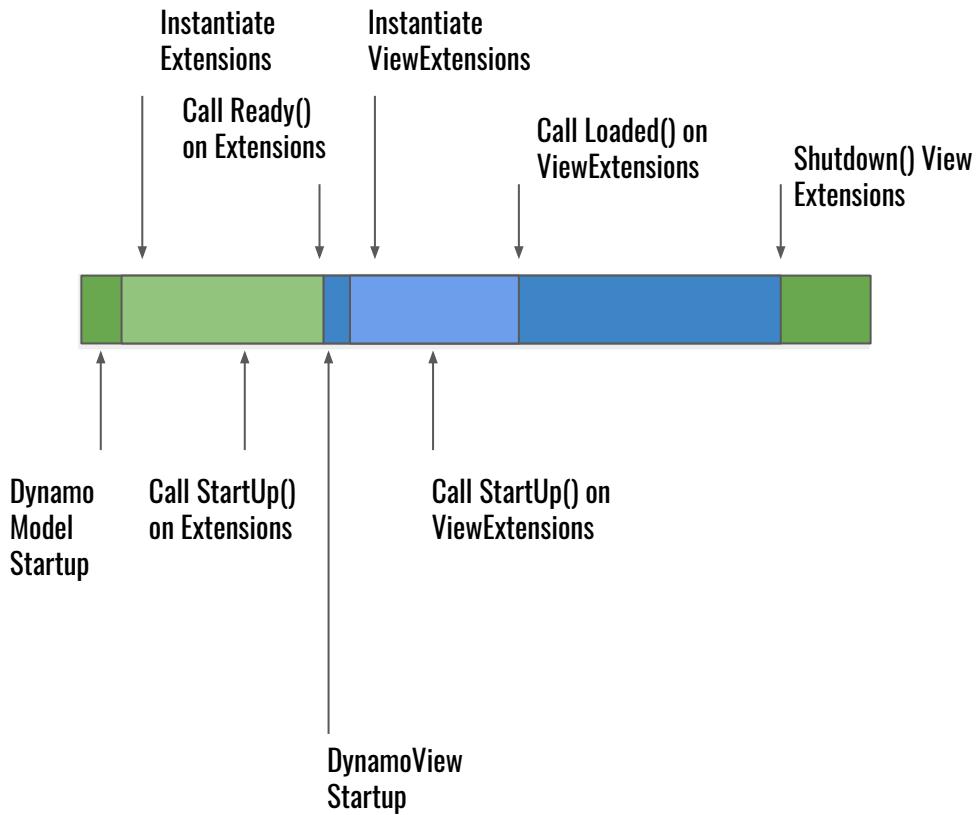


Extensions Architecture

Extension LifeCycle:

StartUp(Params): called when Dynamo starts loading.

Ready/Loaded(Params): called when Dynamo is finished loading.

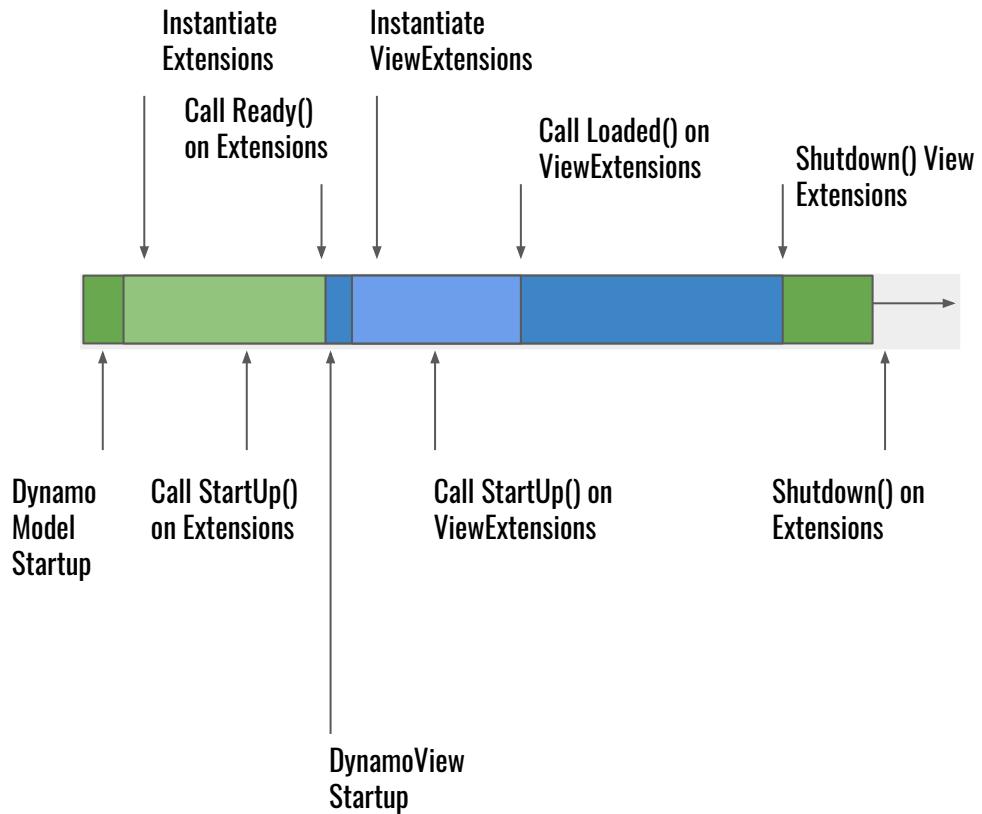


Extensions Architecture

Extension LifeCycle:

StartUp(Params): called when Dynamo starts loading.

Ready/Loaded(Params): called when Dynamo is finished loading.

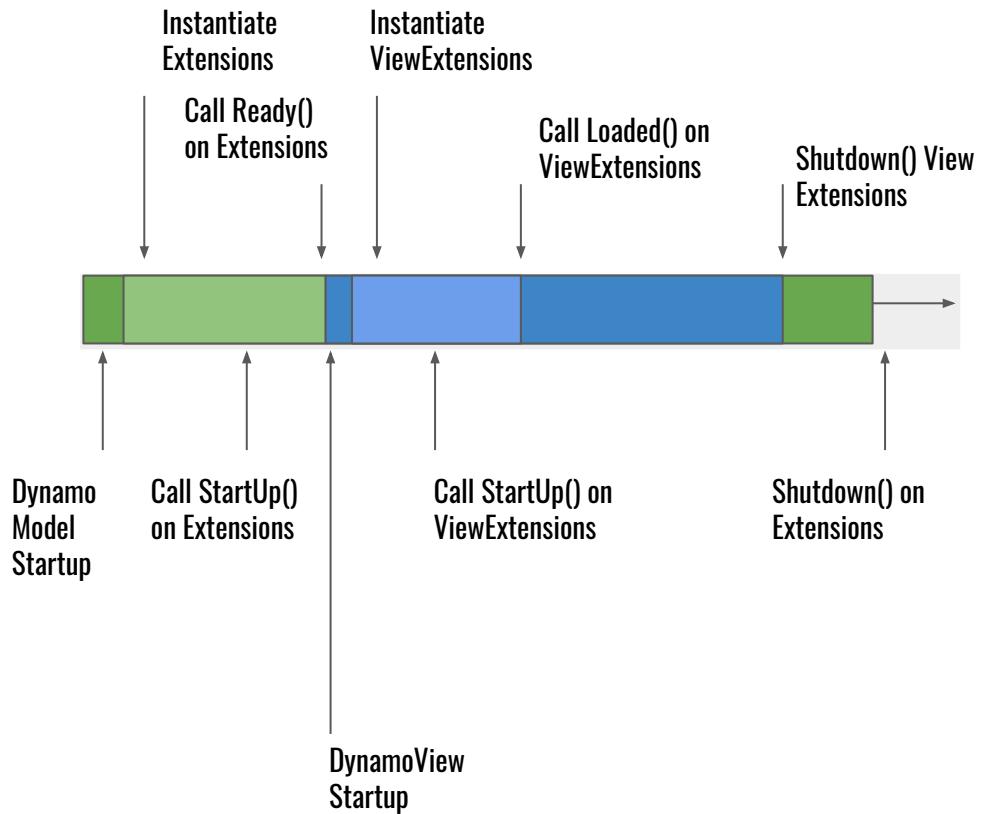


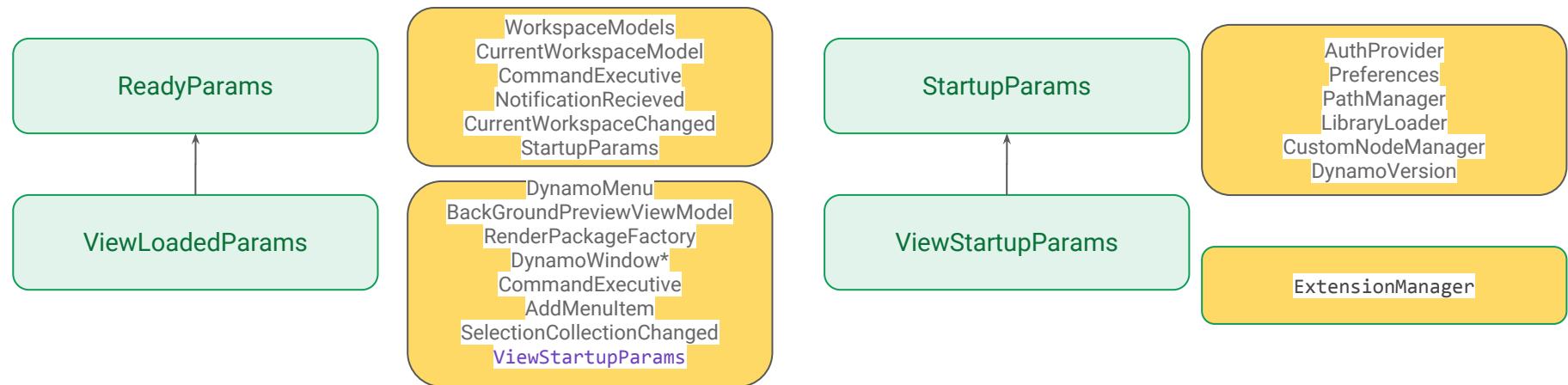
Extensions Architecture



Extension LifeCycle:

Extensions loaded Via Packages will not have *Startup()* Called if they are installed after DynamoStartup!





Making stuff happen in Dynamo (Dynamo APIs)

BackgroundPreviewViewModel

```
public interface IWatch3DViewModel
{
    string PreferenceWatchName { get; }

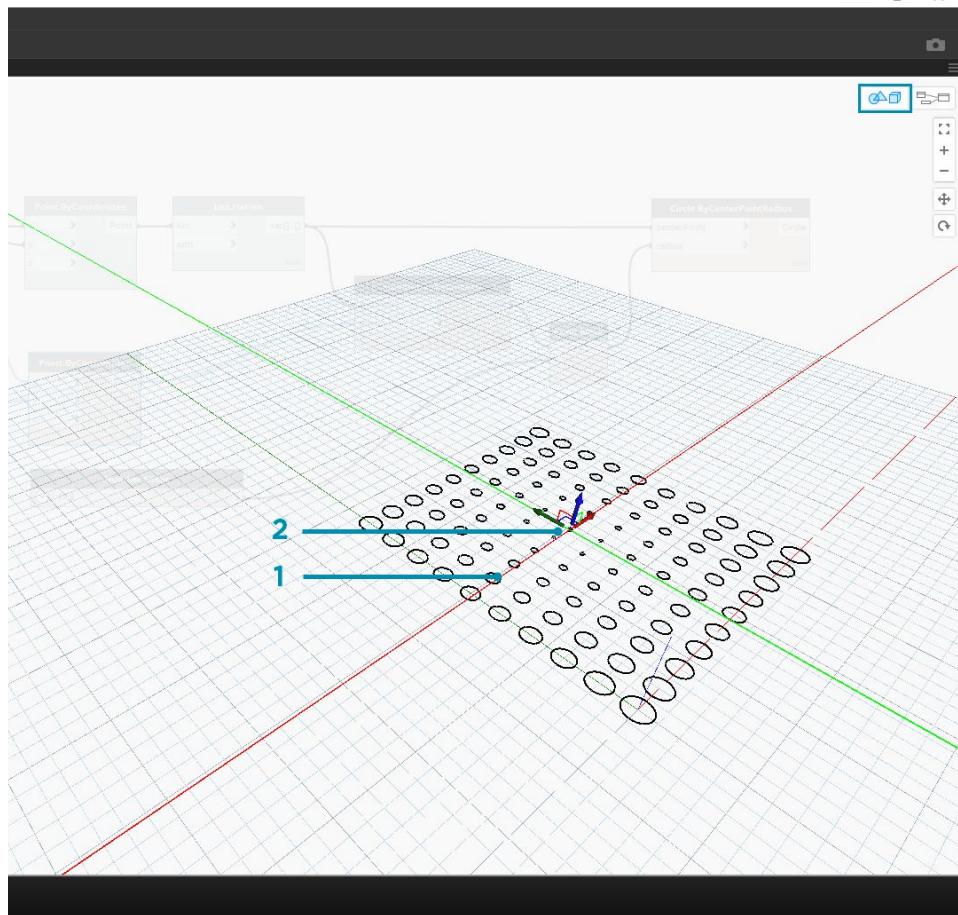
    event Action<object, MouseButtonEventArgs> ViewMouseDown;
    event Action<object, MouseButtonEventArgs> ViewMouseUp;
    event Action<object, MouseEventArgs> ViewMouseMove;
    event Action<bool> CanNavigateBackgroundPropertyChanged;
    event Action<object, RoutedEventArgs> ViewCameraChanged;

    IRay GetClickRay(MouseEventArgs args);

    Point3D? GetCameraPosition();

    CameraData GetCameraInformation();

    void AddGeometryForRenderPackages(RenderPackageCache packages, bool forceAsyncCall = false);
    void DeleteGeometryForIdentifier(string identifier, bool requestUpdate = true);
    void DeleteGeometryForNode(NodeModel node, bool requestUpdate = true);
    void HighlightNodeGraphics(IEnumerable<NodeModel> nodes);
    void UnHighlightNodeGraphics(IEnumerable<NodeModel> nodes);
    void Invoke(Action action);
}
```



ExtensionExample: IExtension

Startup(StartupParams)

Ready(ReadyParams)

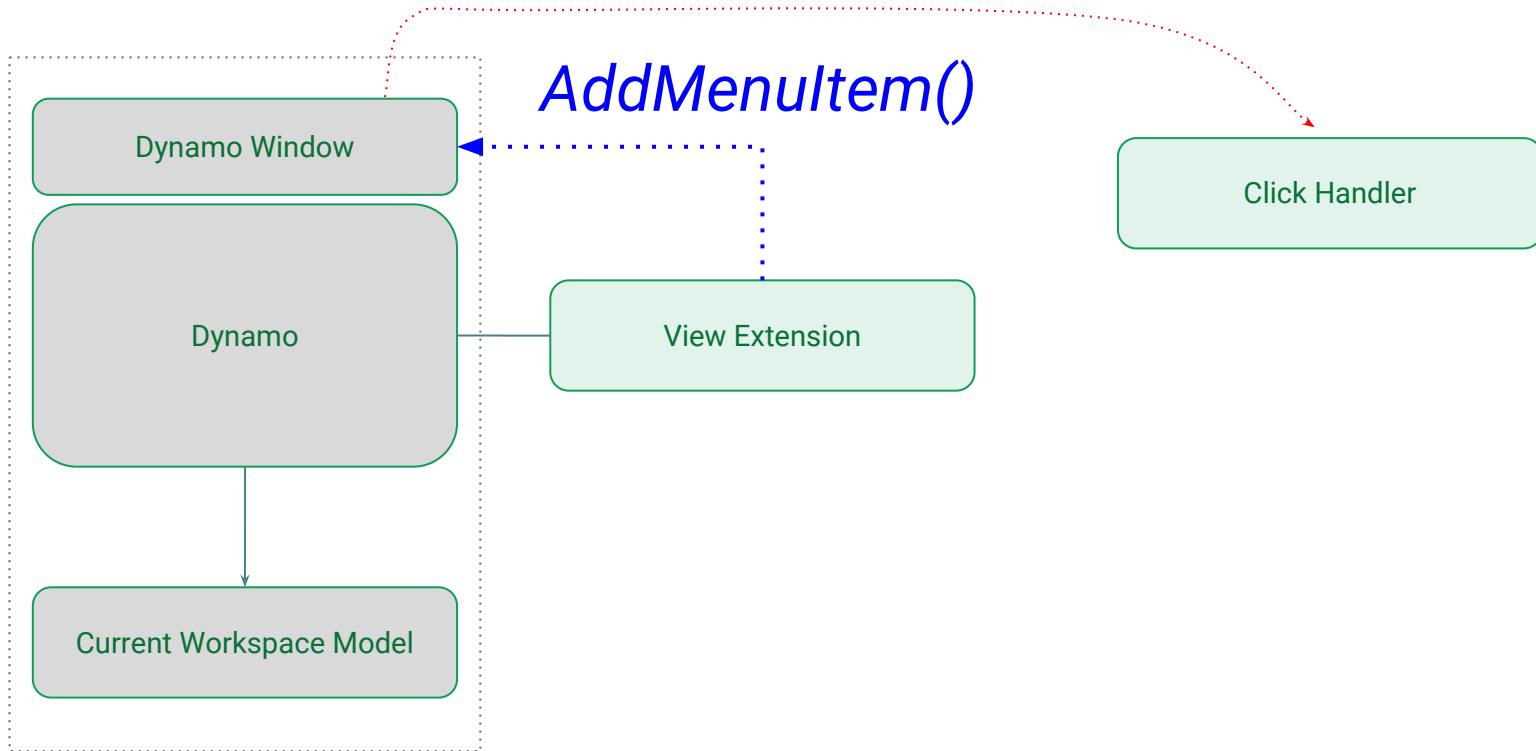
Shutdown()



1. MessageBox.Show("ready")
2. RegisterEventHandlers()

1. UnRegisterEventHandlers()

Click Event



Hello dynamo extension diagram

ViewExtensionExample: IViewExtension

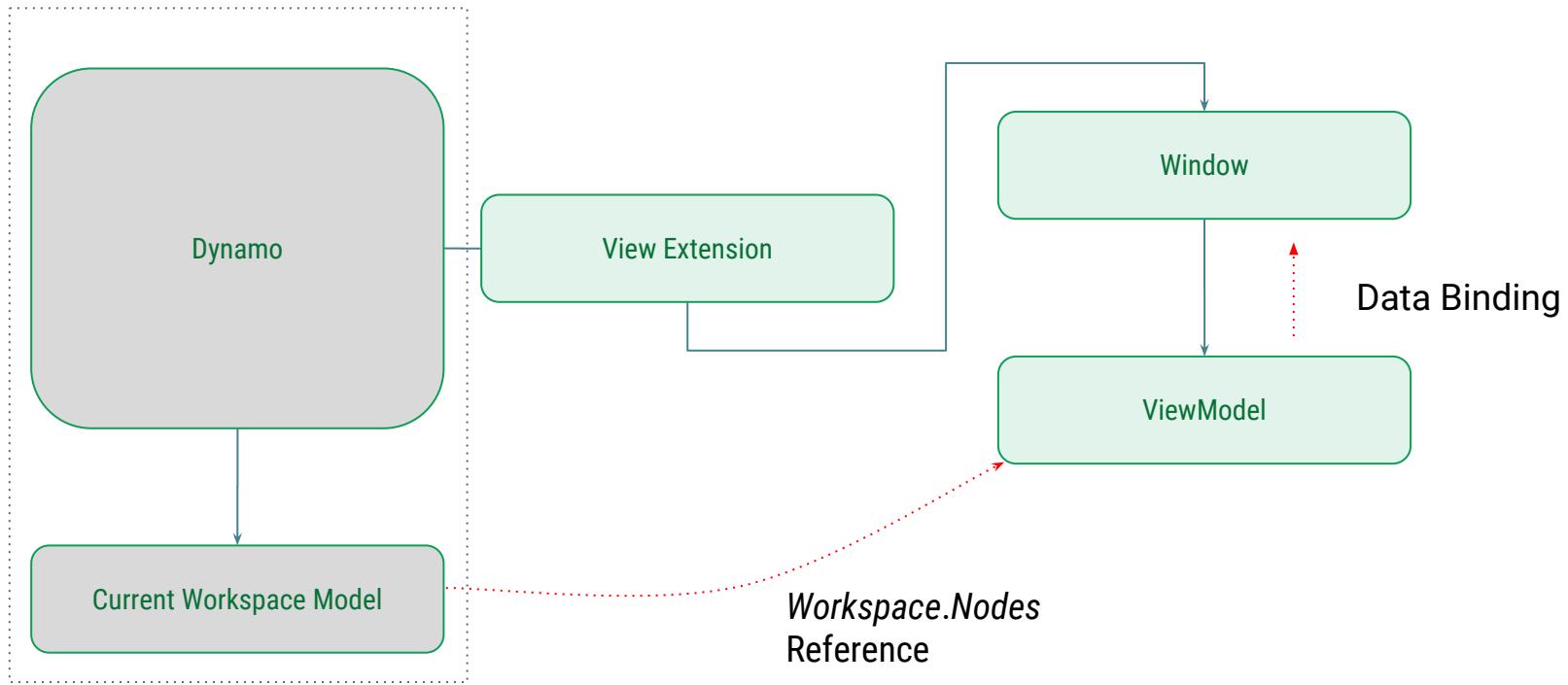
Startup(StartupParams)

Loaded(ViewLoadedParams)

Shutdown()



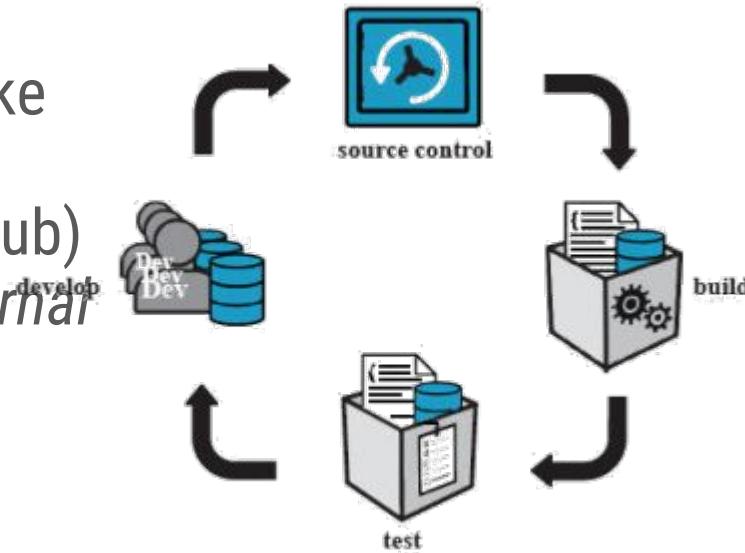
1. MessageBox.Show("ready")
2. MakeMenuItems()



dynamosamples extension diagram

Developing Dynamo

- Managing Source is like worksharing
- One source code (github)
- Multi User (*Some external to Autodesk*)
- Multi Team
- Library and Product



Testing Dynamo

- Regression testing uses “command” files and NUnit testing.
- Dynamo sessions can be recorded and most interactions played back and compared to past runs that should be the same.
- Unit Testing executed as part of build process.
- ‘Clash Detection’? ‘Model Checking’?

