# Cinch Release Notes

This document holds the release notes for Cinch, and details what has changed at each release.

## NOTE:

As previously mentioned now that CinchV2 is out, CinchV1 will no longer receive that much love, CinchV2 is where all the new changes will be made.

## Current Release

Added the following requested features:

1. Modified the WPFOpenFileService/WPFSaveFileService to allow FileName to be set to new value. Previously any set of FileName was ignored. See this Cinch post for details : <http://cinch.codeplex.com/workitem/5300>
2. Added the ability for the WorkSpaceData classes to be populated with their constructed ViewModel, such that the current ViewModel could be used to create a new ViewModel with the same data.

See these Cinch form posts for details : <http://cinch.codeplex.com/Thread/View.aspx?ThreadId=239314> and <http://cinch.codeplex.com/Thread/View.aspx?ThreadId=237461> and <http://www.codeproject.com/KB/WPF/CinchV2_4.aspx?msg=3685737#xx3685737xx>

1. Made the CinchV2.WPFUIVisualizer service use WeakEvents, as there was a potential that the spawning ViewModel could live longer than the popup. I had missed this one, and although I do not think it was a memory leak as such, as it was only using memory while parent VM was around, and that got GC’s when its parent VM was GCd, it was worth fixing, so I fixed it.
2. Made the CinchV2.ChildWindowService unhook from the ViewModelBase. CloseRequest event when the ChildWindow is closing. Which allows the ChildWindow to be GCd
3. One user spotted something silly that I did which was pretty stupid of me, I implemented the IDisposable interface in Cinch.ViewModelBase class. And according to MEF programming guidelines : <http://mef.codeplex.com/wikipage?title=Parts%20Lifetime&referringTitle=Guide> if your part (and Exported ViewModels are MEF parts) and your part implements IDisposable even if you declare your part with a [PartCreationPolicy(CreationPolicy.NonShared)] MEF will keep the part hanging around, as it expects you to ask the CompositionContainer to release it, by using a call to ReleasePart(..). Now as Meffed is not my code, the solution seemed to be to not use the IDisposable interface. So what I have done is to create a new ICinchDisposable interface which has single Dispose() method, and that is what the CinchV2 ViewModelBase class now uses. I should have spotted that one. Sorry

## 52330 Release : 17/10/10 10:54AM Uk time

One user reported DataWrapper<T>.IsDirty was not working in Silverlight even though it was in WPF. The user reported a fix which I have included in SL code branch, I left WPF code alone as it was working just fine. To read more about this you can read this post : http://www.codeproject.com/Messages/3620814/IsDirty-in-a-DataWrapper-bound-to-by-a-checkbox-wi.aspx

## 51807 Release : 02/10/10 10:45AM Uk time

Added the following user suggestions

1. Additional IsReadOnly property on DataWrapper<T> which is the inverse value of IsEditable DataWrapper<T> property, but is also a full INPC property. This means that using the DataWrapper<T> with controls that support IsReadOnly, such as TextBox is even easier, and no longer requires a ValueConverter
2. Added in new ParentViewModel property in DataWrapper<T>. Which can be used to perform cross property validation within a DataWrapper<T> rule. For more information on this, see : <http://cinch.codeplex.com/Thread/View.aspx?ThreadId=228895>
3. Added cleanup method to the IViewAwareStatus/ViewAwareStatus for Silverlight. I would liked to have used WeakReference approach like I did with WPF version, but ran into issue with Silverlight, so opted for a new PerformCleanUp() method that unhooks the FrameworkElements Loaded/Unloaded events (which allows View to be GCd). The best place to call this is in the ViewModel Dispose methods, where you should override the ViewModelBase dispose method. The only tricky bit is figuring out when to call Dispose() on your Disposable ViewModel. The Cinch.SL demos are unable to show this due to the style of demo, but what I would recommend is that if you have some sort of tabbed interface, have a close button on the TabItem which can call Dispose() on your Disposable ViewModel, which will unhook all events for you.

## 51137 Release : 14/09/10 6:59AM Uk time

One CinchV2 user spotted a weirdness where design time data was not working with latest version of MefedMVVM (even though I swore I checked it), I talked to this with Marlon and he did make a breaking change by accident, which has now been fixed in MefedMVVM 60612. So I apologize dear users, but this time it was not me, it was him Marlon. Tee Hee. At least it is all good now, all fixed and verified and checked in. Based on this Marlon and I have agreed that every time he updates MefedMVVM we will test it against CinchV2 even if I decide not to resync with MefedMVVM at that stage.

This release also includes a tiny fix for the debugger display string in the WorkSpaceData.ToString() method. My bad ooops

## 50505 Release : 14/08/10 11:48AM Uk time

Add a couple of enhancements that users have requested, I was waiting until I got a good list of enhancements together and do them all at once, to minimise the number of downloads. So the following have been done:

* Altered Cinch V1 DataWrapper to take a Action in its constructor such that when the Cinch code generator is used the non auto generated part of the overall ViewModel class will be called back using the Action passed to the constructor of the DataWrapper when a property changes. Also got rid of ridiculous condtructors on the DataWrapper class, and also made it a partial class
* Modified the Cinch code generator to support this, so when you next run the code generator you will see what it did. I basically added another callback call inside of DataWrapper<T> class
* Altered Cinch V2 DataWrapper the same way as I did for V1 above.
* Removed small issue in the WorkspaceData.ToString() override where I did not take into account a possible null dataValue. As you never know what this value could be (it could be any object) I have removed the ToString for the dataValue entirely. This is described at : <http://cinch.codeplex.com/workitem/4917>
* Made the following classes partial
  + ValidatingObject (Cinch v1 and Cinch v2)
  + ValidatingViewModelBase (Cinch v1 and Cinch v2)
  + ViewModelBase (Cinch v1)
  + ValidatingViewModelBase (Cinch v1)
  + WorkspaceData (Cinch v2 WPF only)
* Added a new ViewAwareStatusWindow service targeting Windows
* Added a Non mandatory extra event to WorkSpaceData that can be used to cancel a closing of a workspace. Revised WPF demo to show this for ImageLoader workspace and left About workspace as it was to show it is not mandatory.
* Resynced with [MeffedMVVM release 58837](http://mefedmvvm.codeplex.com/SourceControl/changeset/changes/58837)

## 48912 Release : 19/07/2010 8:27PM

Should all be good, with CinchV1 and CinchV2, note that CinchV2 demo apps are zipped.

**CinchV2 demo instructions**

You will need to follow these instructions to get CinchV2 demo apps working

1. Open the CinchV2.sln solution, and unload the CinchV2 project in VS2010
2. Compile the CinchV2.WPF and CinchV2.SL projects within the CinchV2.sln solution
3. Unzip the SL4\_Demo.zip project, open SL4\_Demo.sln solution and then do the following :
   1. Within the CinchV2DemoSL project remove the references for CinchV2.SL.dll and MEFedMVVM.SL.dll
   2. Copy Cinch.SL.dll from CinchV2.SL\Bin\debug to SL4\_Demo\CinchV2DemoSL\Lib
   3. Copy MEFedMVVM.SL.dll from CinchV2.SL\Bin\debug to SL4\_Demo\CinchV2DemoSL\Lib
   4. Now for CinchV2DemoSL demo project reference Cinch.SL.dll and MEFedMVVM.SL.dll from SL4\_Demo\CinchV2DemoSL\Lib
   5. With the DesignTimeServicesSL project remove the references for MEFedMVVM.SL.dll
   6. Copy MEFedMVVM.SL.dll from CinchV2.SL Bin\debug to SL4\_Demo\DesignTimeServicesSL\Lib
   7. Now for DesignTimeServicesSL project reference MEFedMVVM.SL.dll from SL4\_Demo\ DesignTimeServicesSL\Lib
   8. Compile the open SL4\_Demo.sln solution
4. Unzip the WPF\_Demo.zip project, open WPF\_Demo.sln solution and then do the following
   1. Within the CinchV2DemoWPF project remove the references for CinchV2.WPF.dll and MEFedMVVM.WPF.dll
   2. Copy Cinch.WPF.dll from CinchV2.WPF\Bin\debug to WPF\_Demo\CinchV2DemoWPF\Lib
   3. Copy MEFedMVVM.WPF.dll from CinchV2.WPF\Bin\debug to WPF\_Demo\CinchV2DemoWPF\Lib
   4. Now for CinchV2DemoWPF demo project reference Cinch.WPF.dll and MEFedMVVM.WPF.dll from WPF\_Demo\CinchV2DemoWPF\Lib
   5. With the DesignTimeServicesWPF project remove the references for MEFedMVVM.WPF.dll
   6. Copy MEFedMVVM.WPF.dll from CinchV2.WPF Bin\debug to WPF\_Demo\DesignTimeServicesWPF\Lib
   7. Now for DesignTimeServicesWPF project reference MEFedMVVM.WPF.dll from WPF\_Demo\ DesignTimeServicesWPF\Lib
   8. Compile the open WPF\_Demo.sln solution

## 48912 Release : 19/07/10 8:27PM Uk time

SVN checkin missed a few files, fixed that. Grr

## 48661 Release : 17/07/10 9:36AM Uk time

Realised that the fix that I checked in to fix WPF build caused a problem with SL demo, which stupidly I did not check. Yes I am an idiot. Anyway the net result is that I had to fix up an issue that was affecting SL demo and Cinch.SL. This is now all done and tested, and I also took the opportunity to resynch Cinch with MeffedMVVM release : 55859, 03/07/2010

I think this will be the last of my changes, I am sorry that there have been a few checkins lately, but there were things that I had to change, sorry about that.

## 48484 Release : 14/07/10 9:33PM Uk time

Forgot to add a new file.GRRR

## 48481 Release : 14/07/10 8:41PM Uk time

One user alerted to me to a possible issue with the way the WPF workspaces work. This led me to look into a possible memory leak area, I talked to Marlon Grech about this, and it turned out to be a problem with way we were both doing StatusAware services. This has now been fixed up with WeakReferences as well as the CommandCompleted event in SimpleCommand<T1,T2> which also caused a strong reference of the View to be maintained. This is all fixed and tested.

## 48252 Release : 11/07/10 10:32AM Uk time

Added PartCreationPolicy MEF attributes to ViewModels in WPF/SL demo projects (as it was confusing to some users to always be getting the same ViewModels (although that is ok for the demos, its not very real world like, so showed a more real world scenario, where each View would get a new instance of a ViewModel which is achieved using [PartCreationPolicy(CreationPolicy.NonShared)].

Also to enable this, I have included a specialised TabControl for WPF which allows all open Views to be stored in memory and just shows/hides active one. The standard TabControl in WPF works different to this where is creates entire VisualTree again, so that would be a disaster using MefedMVVM as that would mean we would get a new VM each time we change tabs. So the WPF demo shows you how to work around this using a special TabControlEx which has a code file and Style (AppStyles.xaml) that you can use. See the WPF demo app for that.

As for SL, as the TabControl in SL lacks the correct overrides, I am afraid to say you are on your own in trying to figure out that one.

## 48188 Release : 08/07/10 09:15AM Uk time

Codeplex auto checkin due to server upgrade

## 47479 Release : 19/06/10 08:14AM Uk time

Adding missing ViewModelBase IsCloseable/DisplayName for compatibility with CinchV1s way of managing workspaces, even though CinchV2 is far far better approach

## 47438 Release : 17/06/10 11:56AM Uk time

## Fixed problem with CanExecute and WeakReferences and updated demos accordingly

## 47316 Release : 14/06/10 10:20PM Uk time

## Fixed small design time issue with MainWindowViewModel workspaces

## 47270 Release : 13/06/10 19:20PM Uk time

Good checkin of CinchV2

## 47259-47269 Release : 13/06/10 17:33PM Uk time

Failed SVN checkins for CinchV2

## 46208 : Release : 08/05/10 09:35AM Uk time

Stable checkin for CinchV2

## 46191-46207 Release : 08/05/10 09:26AM Uk time

Fixed SVN repository, and fixed IUIVisualizer issue with CinchCodeGen

## 46191-46207 Release : 08/05/10 Various times

Corrupt SVN repository, attempting to fix

## Current Release : 04/05/2010 09:47AM Uk time

Applied all patches that I deemed good

## 46032 Release : 04/05/2010 09:07AM Uk time

Updated release notes to show new usage for MediatorMessageSinkAttribute

## 46013 Release : 03/05/2010 10:35PM Uk time

These are the things that were fixed with this release

| **Area** | **Notes** |
| --- | --- |
| Put in new Mediator, and make ViewModel unregister on Dispose | There has been a lot of work done in the Mediator which is now a Singleton, which can be used within Views also, and also extra Unregister/Register methods have been made available, as well as Async methods.  The ViewModelBase also unregisters within the Dispose() method. |
| **Cinch Issues:**  Mediator in ViewModelBase better be static |
| **Cinch Discussions:**  Mediator Message Method Signatures |
| **Cinch Issues:**  NPE in WPFUIVisualizerService | Done |
| **Cinch Issues:**  DoubleclickData is internal and cannot be used | Made the DoubleClickData class public. |
| **Cinch Discussions:**  Important issue to solve (Unity) | Have made a default UnityProvider which Cinch uses unless new one is set on constructor of ViewModels. |
| **Cinch Discussions:**  Adding IsActivated property | Done, but will only work providing inheritors of Cinch ViewModelBase classes, call base.OnWindowLoaded() etc etc |
| **Cinch Discussions:**  DataWrapper and DataValue change notifications, this should work now | These have worked for ages. Working as expected. |
| **Cinch Discussions:**  IsDirty DataWrapper<T> firing INPC event | Done |
| Remove dependency on SLF | Have made a default SLF logger which Cinch uses unless new one is injected. |

**Breaking changes:**

1. The way you register popups is now different, see MainWindow.xaml.cs for details
2. MediatorMessageSinkAttribute no longer needs Type Parameter, it only needs 1 value specified in the attribute parameters.

## 46009 and Older Releases

Please see codeplex releases tab, and comments