**Architecture Overview**

Infrastructure layer provide the following modules

1. Logging Module
2. Exception Handling Module
3. Validation Module

**Target**

The main target of this layer is that grouping all required modules that will be used through all application layers including Client side and Server side layers.

**Problems**

* The main problem is that our project is a SilverLight project, so we need to consider that we will use 2 different kind of .Net applications which are
  + Server side application that uses the full functionality of .Net Framework4.
  + Client side [SilverLight] application that uses some of .Net Framework 4.
* We need our modules to be applicable on both sides, Client Side and Server Side.

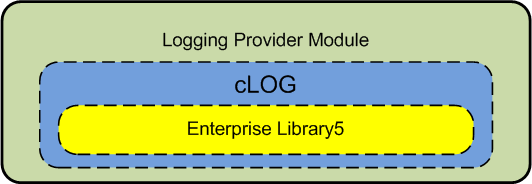
**How To**

* To solve the first problem then we need to split our Infrastructure layer into 3 Main Applications which are
  + citPOINT.eNeg.Infrastructure.Common 🡺 which is a silver light library application.
  + citPOINT.eNeg.Infrastructure.ClientSide🡺which is a SilverLight Library application.
  + citPOINT.eNeg.Infrastructure.ServerSide🡺which is a normal class library application.
* To solve the second problem we will use other third party components like **cLOG.**

**LOGGING MODULE**

**Architecture Overview**

In this module we will wrap the third party component “cLOG” that enables us to make logging on both sides which are client side and server side and in the same time this third party component itself wraps the logging block of enterprise library 5 as the internal strategy for it to allow logging operation

****

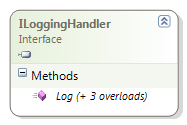
This module will contains the following components

***Enums:***

1. Category enum
   1. Infrastructure.
   2. ServerData.
   3. Web.
   4. ClientData.
   5. Model.
   6. ViewModel.
   7. Common.
   8. Client.
2. EventType enum
   1. Error
   2. Information
   3. Warning
3. Priority enum
   1. Lowest
   2. Low
   3. Normal
   4. High
   5. Highest

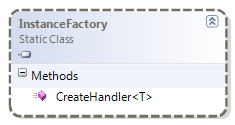
***Interfaces***

1. ILoggingHandler

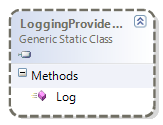


***Classes***

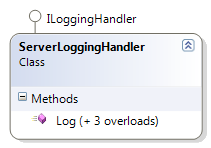
1. InstanceFactory



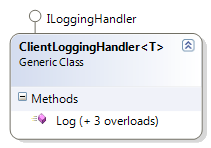
1. LoggingProvider



1. ServerLoggingHandler

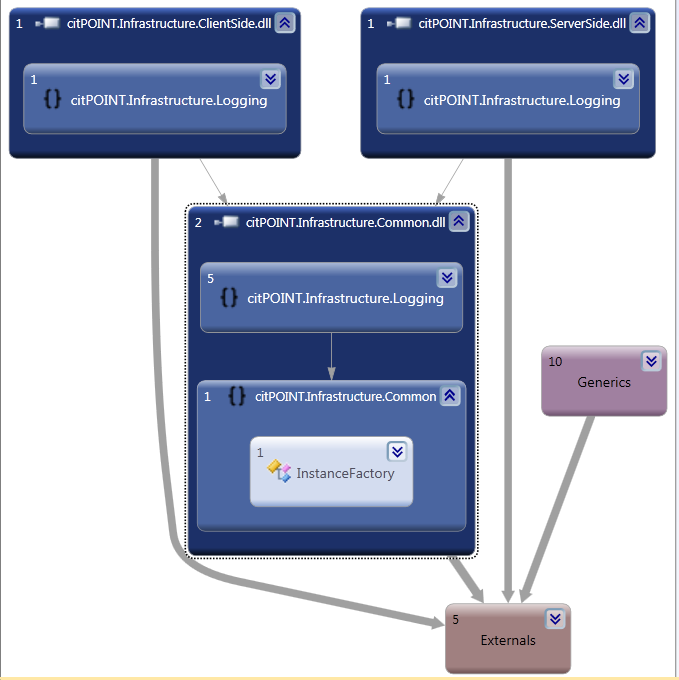


1. ClientLoggingHandler



**Detailed View**

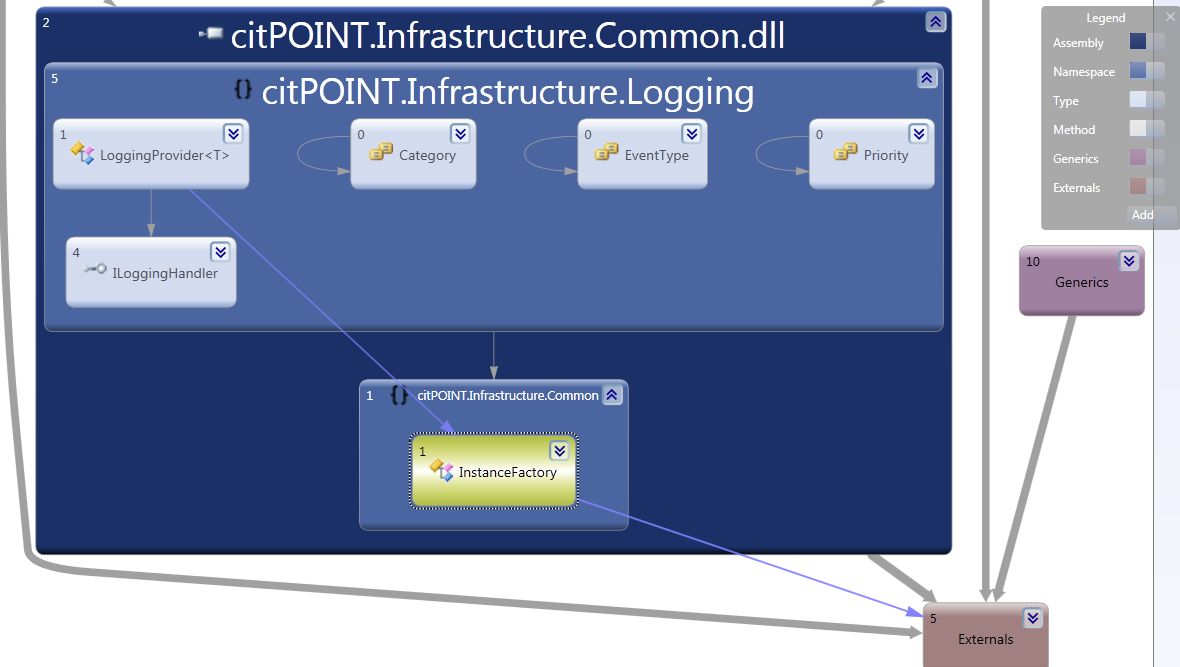
As mentioned in point [How To] we said that we will split Infrastructure layer into three applications, so according for that we will start our implementation for Logging Module like the following



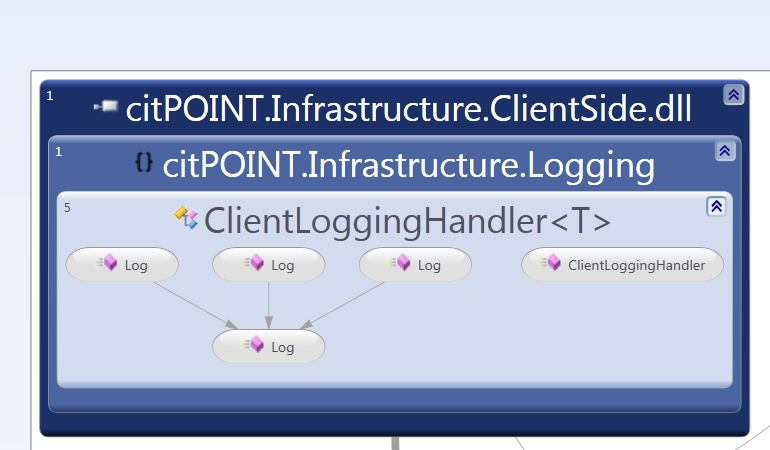
In this figure we can see all assemblies that are participated in Logging module and also notice that the namespace ***citPOINT.Infrastructure.Logging*** will be split into 3 assemblies as you can see in the above figure.

The following figures will show each one of those assemblies in details

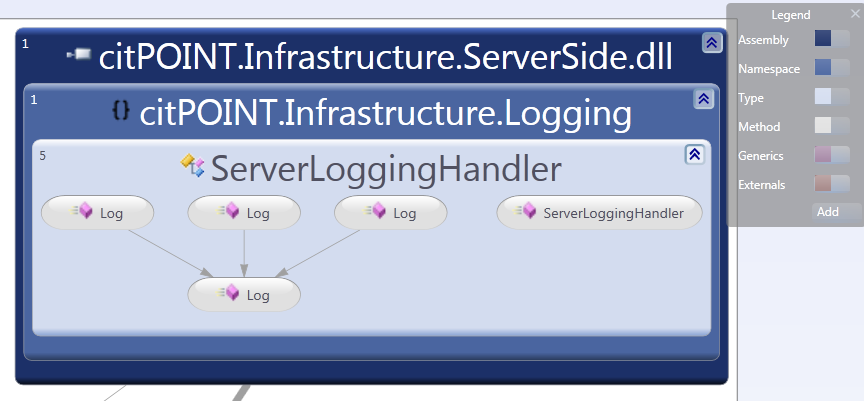
1. citPOINT.Infrastructure.Common



1. citPOINT.Infrastructure.ClientSide



1. citPOINT.Infrastructure.ServerSide



**EXCEPTION HANDLING MODULE**

**Architecture Overview**

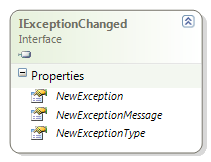
This module will be used to handle exceptions on sides, client side and server side, so also this module will be split into 3 assemblies with the namespace “citPOINT.Infrastructure.ExceptionHandling”.

We will use Enterprise Library 5 to handle exception on server side and on client side we will redirect to the error page and send the error to the server through RIA service and redirect to error page this in case if the client is online but if the client is offline then we will log the exception to isolated storage till the client be online again we will invoke the exception service to log all cached exceptions to the server then clear all cached exceptions.

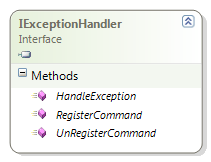
This module will contain the following components

**Interfaces**

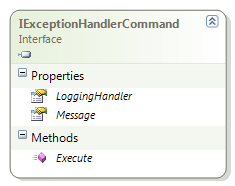
* + 1. IExceptionChanged



* + 1. IExceptionHandler

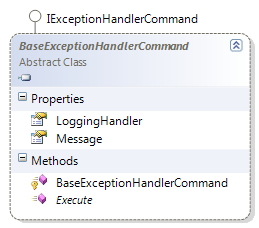


* + 1. IExceptionHandlerCommand

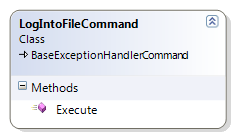


**Exception Command Classes**

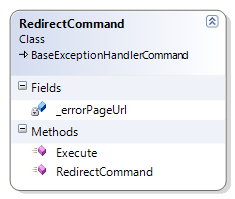
1. BaseExceptionHandlerCommand



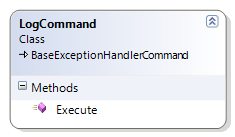
1. LogIntoFileCommand



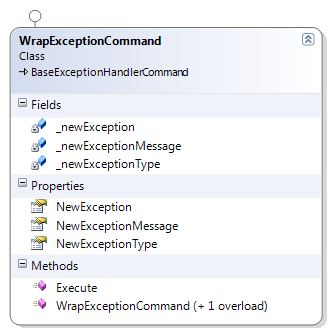
1. RedirectCommand



1. LogCommand

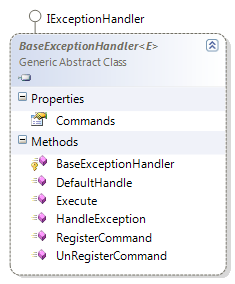


1. WrapExceptionCommand

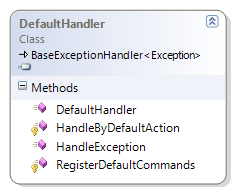


**Exception Handler Classes**

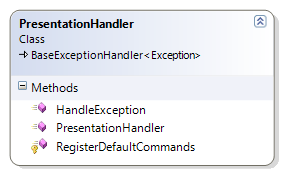
1. BaseExceptionHandler



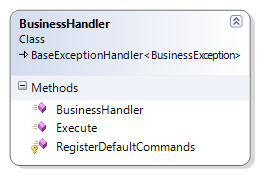
1. DefaultHandler



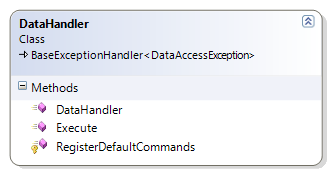
1. PresentationHandler



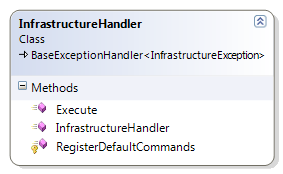
1. BusinessHandler



1. DataHandler

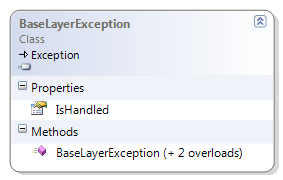


1. InfrastructureHandler

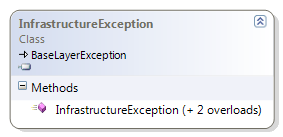


**Custom Exception Classes**

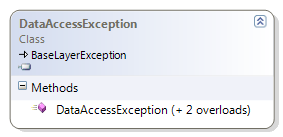
1. BaseLayerException



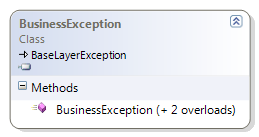
1. InfrastructureException



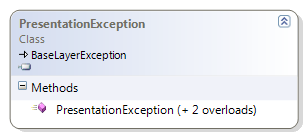
1. DataAccessException



1. BusinessException

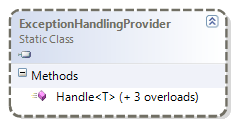


1. PresentationException

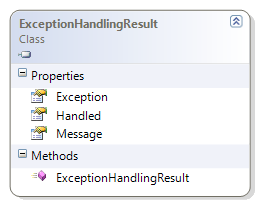


**Main Exception Handling Class**

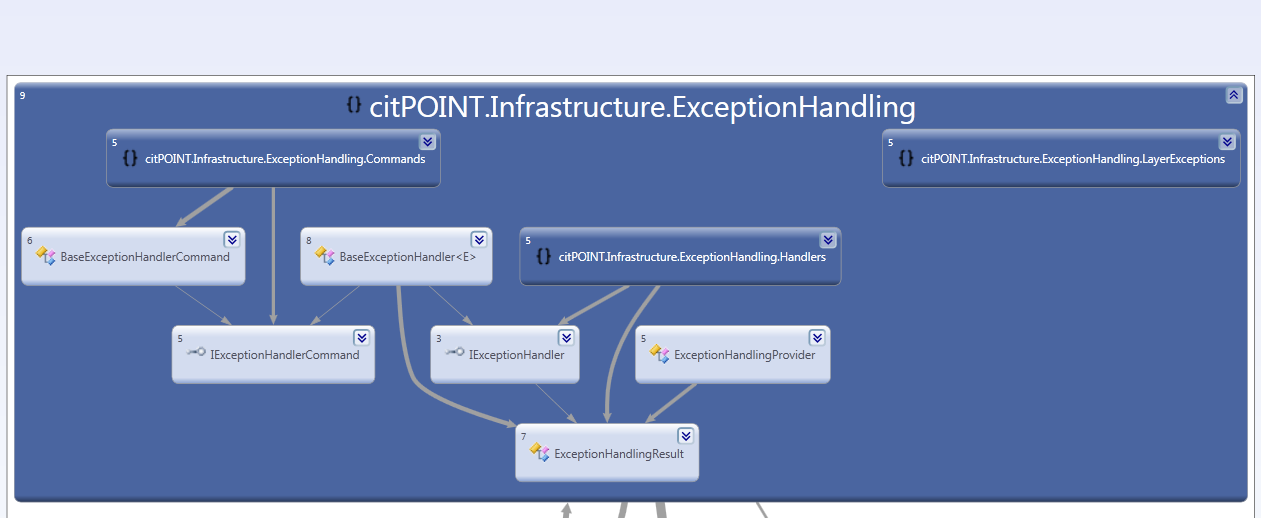
1. ExceptionHandlingProvider



1. ExceptionHandlingResult

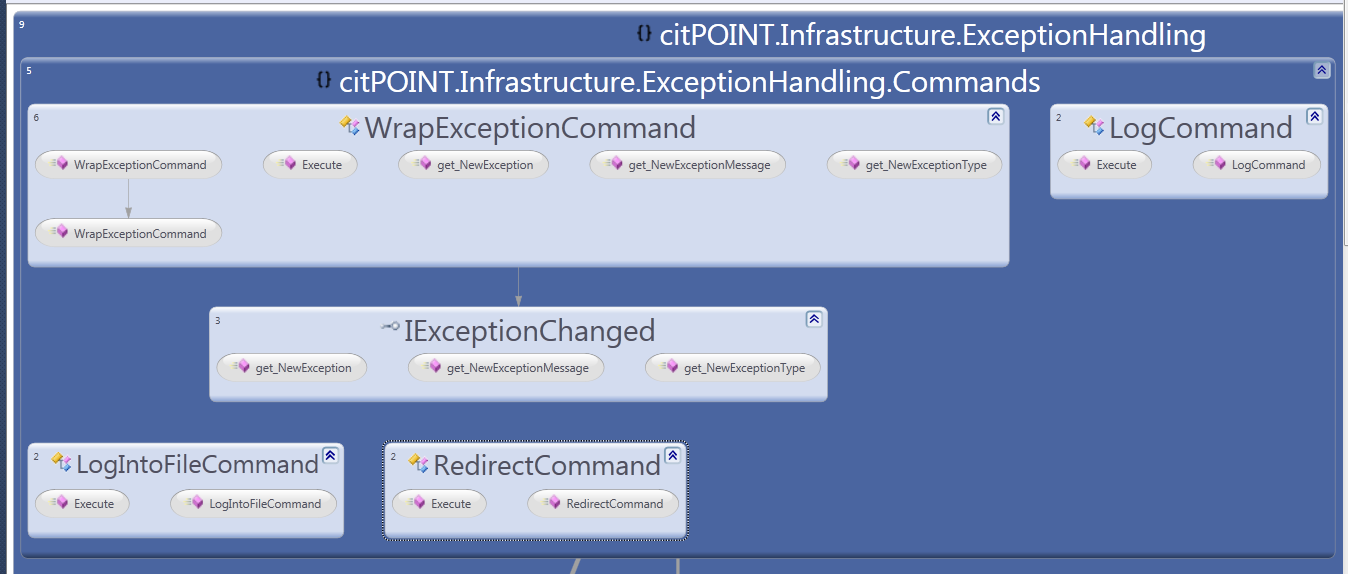


**NameSpace Mapping**

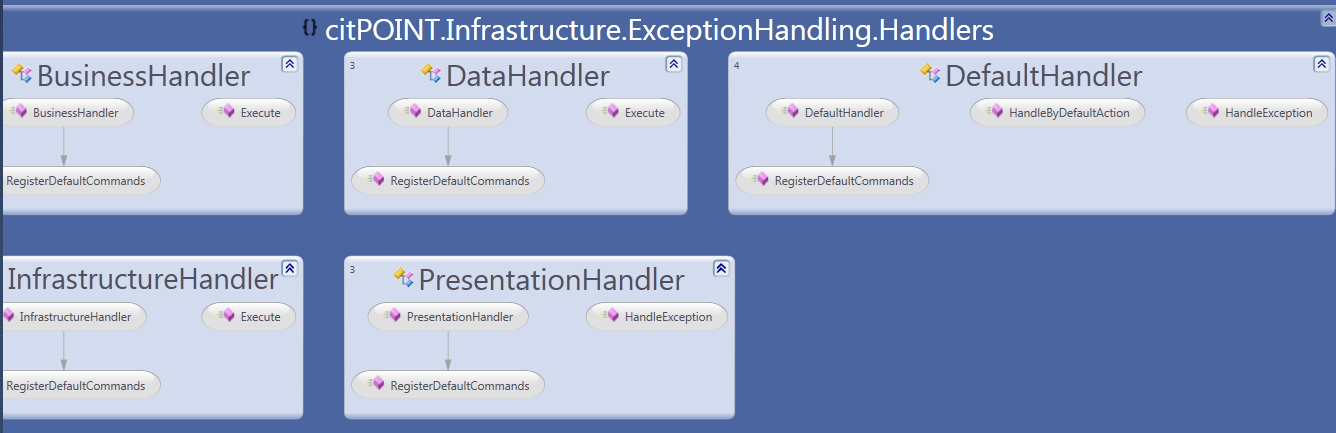
****

**NameSpaces Contents**

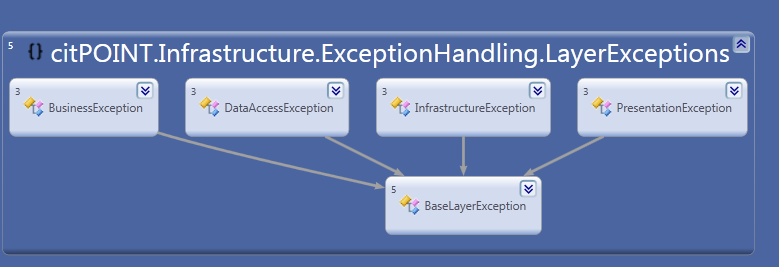
1. **citPOINt.Infrastructure.ExceptionHandling.Commands**



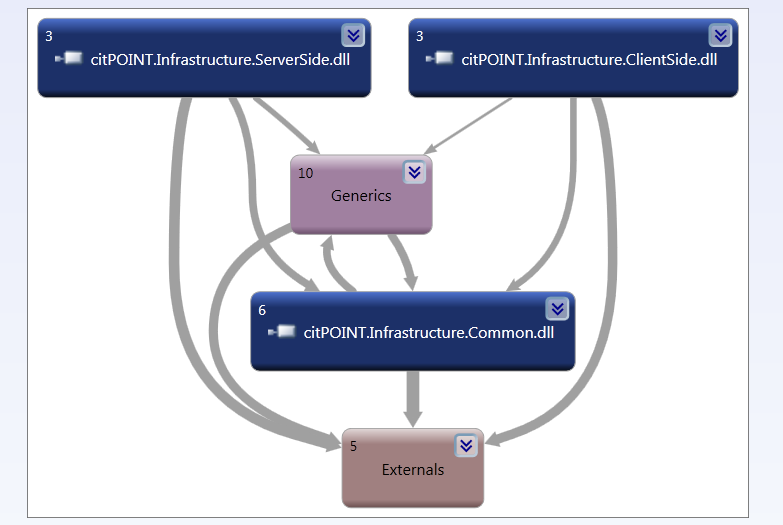
1. **citPOINT.Infrastructure.ExceptionHandling.Handlers**

****

1. **citPOINT.Infrastructure.ExceptionHandling.LayerExceptions**

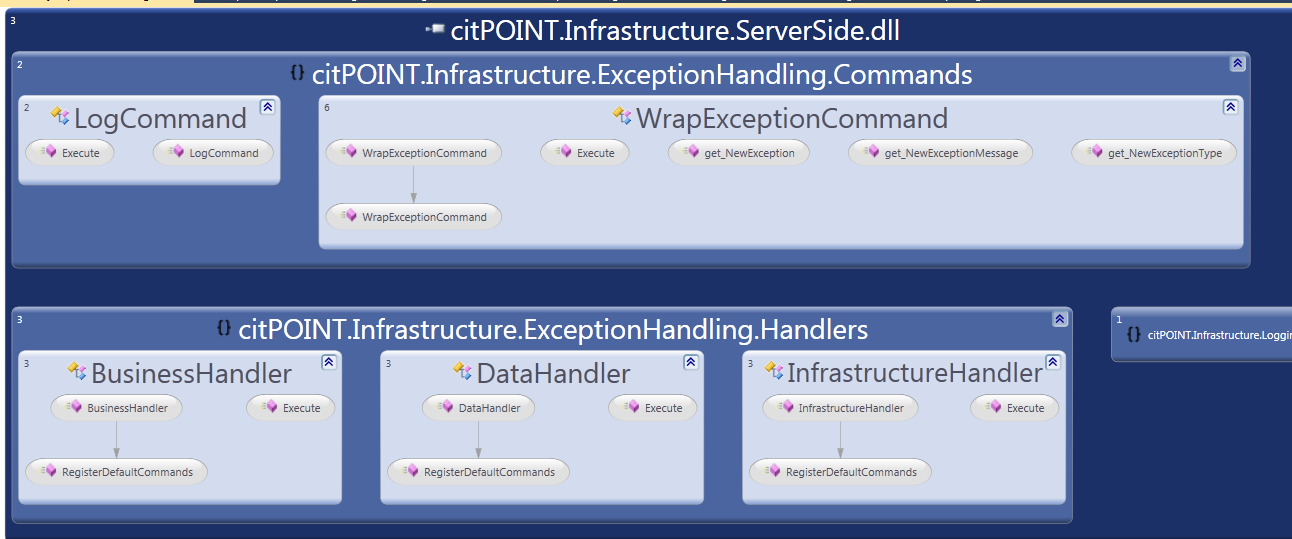


**Assembly Mapping**

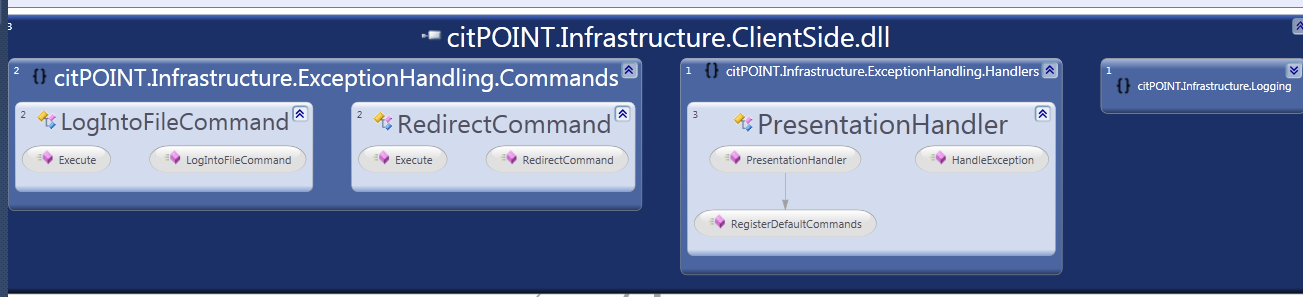


**Assembly Contents**

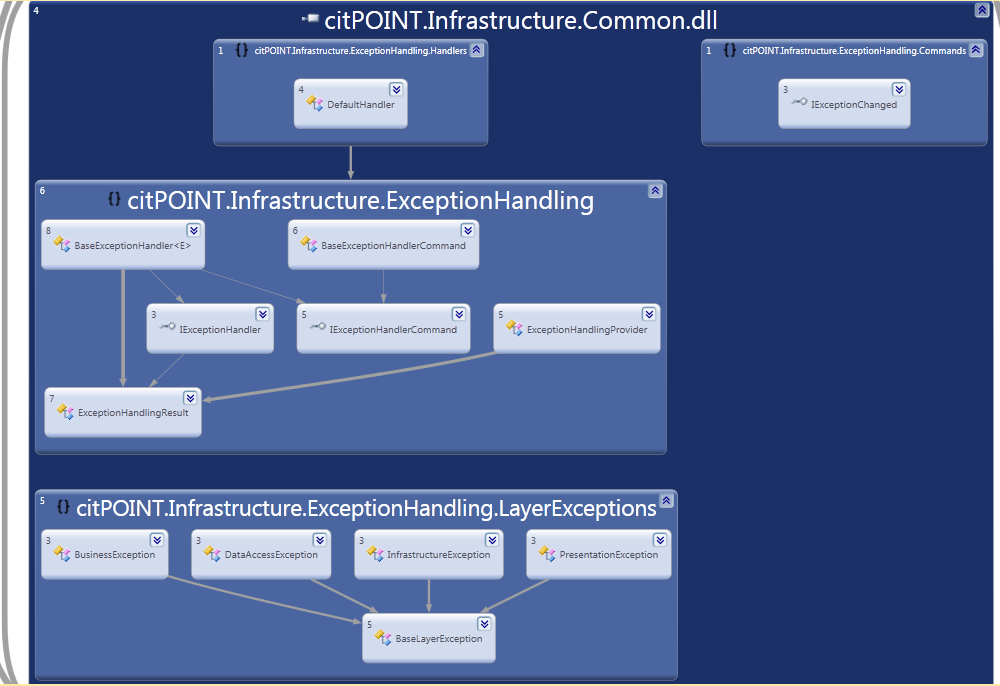
1. **citPOINT.Infrastructure.ServerSide.dll.**

****

1. **citPOINT.Infrastructure.ClientSide.dll**



1. **citPOINT.Infrastructure.Common.dll**

****