

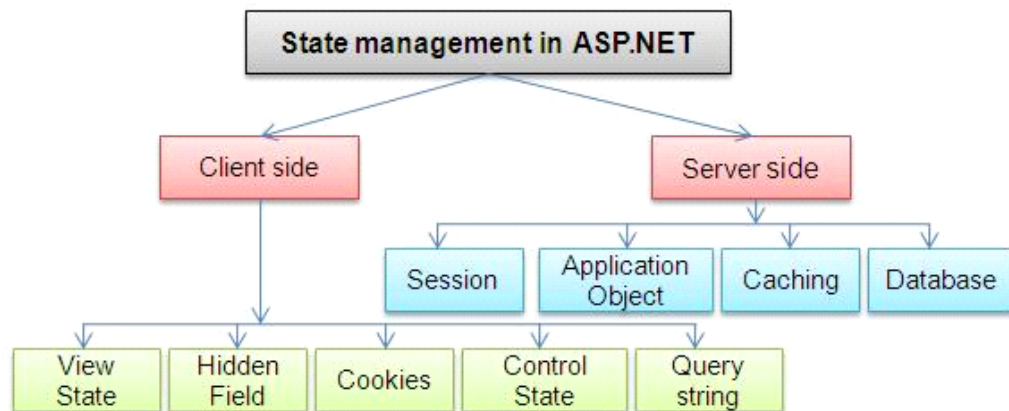
## PageStatePersister를 이용한 ViewState 부하절감

리얼웹	개발본부	배성혁	2010-10-04 오후 1:47
-----	------	-----	--------------------

웹 폼에서 ViewState는 필요악입니다. Stateless인 HTTP 기반의 웹에서, 웹 페이지를 일반적인 프로그램처럼 보이게 하기 위해 ASP에서는 Hidden 필드를 사용했었고, ASP.NET에서는 이를 발전시켜 ViewState를 사용하게 되었습니다.

당연히 Stateless인 HTTP 기반의 웹에 대해, 항상 Stateless이면 좋겠지만, 어차피 기존에도 Hidden Field를 이용한 State를 유지해왔었으므로, 벗어날 수는 없습니다. 다만, 최소화하여 사용하도록 노력해야 합니다만, 이 놈의 ViewState가 아무 생각 없는 개발자들 때문에, 속도 저하의 원흉이 되었다는 것이지요. 양식 있는 개발자라면, 꼭 필요한 ViewState만 사용하고, ViewState 크기를 최소화하여, Page Loading 속도 및 PostBack 시에 요청 속도를 증가 시킬 것입니다.

MS에서 도저히 안되겠는지, ASP.NET 2.0부터는 PageStatePersister라는 클래스를 통해, ViewState를 Page의 Hidden Field 가 아닌 다른 곳에도 저장/로드 할 수 있도록 하는 매커니즘을 제공합니다.



출처: [ViewState: Various ways to reduce performance overhead](#)

즉 기존 Page의 ViewState를 관리하는 기능을 추가하는 것이 아니라, PageAdapter 를 이용하여, ViewState 관리를 다른 방법으로 관리할 수 있도록 쉽게 변환할 수 있습니다.

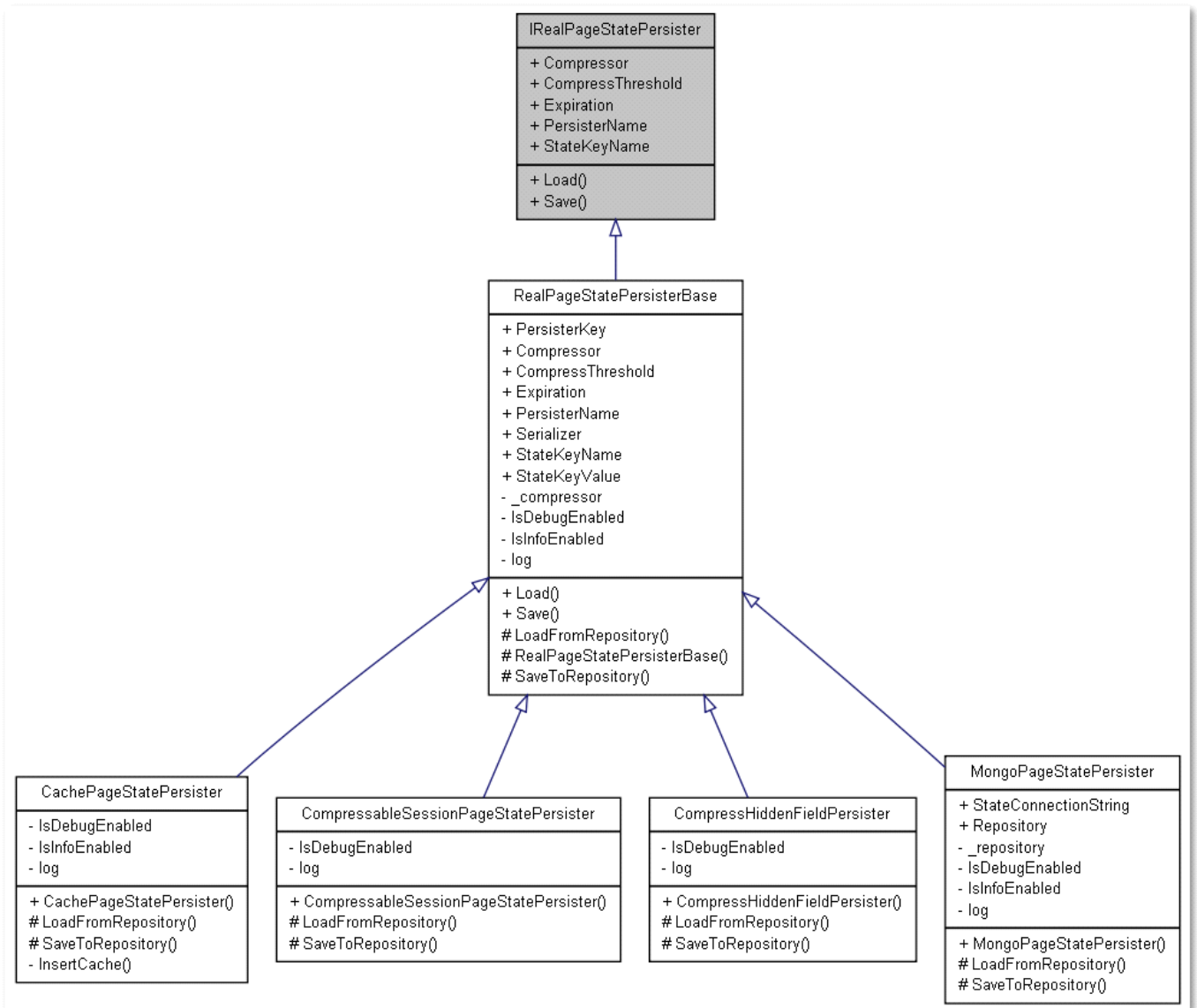
RCL에서는 IoCPageStateAdapter 라는 클래스를 통해, IoC 환경설정에 지정된 IRealPageStatePersister를 Resolve하여, ViewState를 저장소에 저장하고, 필요할 때 로드하여 사용하도록 합니다.

특히 Server Side의 저장소로는 Session, Cache, MongoDB를 사용하도록 하였고, 더 다양한 저장소를 사용하는 사용자 Persister를 쉽게 등록하여 사용할 수 있도록 하였습니다.

이렇게 함으로서, Client 브라우저와 서버간의 통신상에서 데이터 양 중 10~40%를 차지하는 ViewState 정보를 Client에 내려보내지 않고, 또 PostBack으로 다시 받지 않아도, 상태를 유지할 수 있게 되었습니다. 이러한 통신 상의 데이터양의 크기 절감은 특히 네트워크 환경이 열악한 곳에서 더욱 진가를 발휘합니다.

RCL에서는 IRealPageStatePersister, RealPageStatePersisterBase를 기반으로, 다양한 저장소 별로 Persister를 만들었습니다.

기본적으로 모든 Persister는 압축을 지원하므로, 서버상의 자원도 절약합니다. Client에 상태 정보가 전달되지 않으므로, 보안 상의 문제도 자동으로 해결된다고 보시면 됩니다.



IRealPageStatePersister 의 상속 체계

웹 응용프로그램에서 적용하기 위해서는 두가지를 설정해야 합니다.

1. IoCPageStateAdapter 등록하기
  - a. 웹 응용프로그램에 App\_Browsers 라는 폴더를 생성합니다.
  - b. Default.browser 파일을 만듭니다.
  - c. 아래 코드와 같이 Adapter를 등록합니다. (IoCPageStateAdapter)

```

1 <!--
2   다음 위치에서 기존 브라우저 정의를 찾을 수 있습니다.
3   <windir>\Microsoft.NET\Framework\<ver>\CONFIG\Browsers
4   -->
5 <browsers>
6   <browser refID="Default">
7     <controlAdapters>
8       <!-- ViewState 정보를 특정 Repository에 저장할 수 있도록 하는 IRealPageStatePersister를 IoC를 통해 제공하는 Adapter 입니다. -->
9       <adapter controlType="System.Web.UI.Page" adapterType="RCL.Core.IoCPageStateAdapter" />
10
11       <!-- AspImageControlAdapter는 ImageUrl의 상대경로를 절대경로로 변경합니다 : 절대경로를 사용하면 Page Loading 시간을 단축할 수 있습니다.-->
12       <adapter controlType="System.Web.UI.WebControls.Image" adapterType="RCL.Core.AspImageControlAdapter" />
13     </controlAdapters>
14     <capabilities>
15       <capability name="xml" value="true" />
16     </capabilities>
17   </browser>
18 </browsers>
  
```

1. IoC 환경설정에서 IRealPageStatePersister 등록하기

a. 주의할 점 : 모든 Persister의 lifestyle은 Transient 여야 합니다.

```
1  <?xml version="1.0" encoding="utf-8"?>
2
3  <configuration>
4  <!-- PageStatePersister Configuration for IoC / DI with Castle.Windsor -->
5  <components>
6
7      <!-- ASP.NET Cache에 ViewState를 저장하는 Persister입니다.-->
8      <component id="PageStatePersister.Cache"
9          service="RCL.Core.IRealPageStatePersister, RCL.Core"
10         type="RCL.Core.CachePageStatePersister, RCL.Core"
11         lifestyle="Transient">
12         <parameters>
13             <!-- 최소 압축 크기 (단위: byte)-->
14             <CompressThreshold>1024</CompressThreshold>
15             <!-- 정보의 유효기간 (단위: Minute)-->
16             <Expiration>1440</Expiration>
17         </parameters>
18     </component>
19
20     <component id="PageStatePersister.MongoDB"
21         service="RCL.Core.IRealPageStatePersister, RCL.Core"
22         type="RCL.Mongo.MongoPageStatePersister, RCL.Mongo"
23         lifestyle="Transient">
24         <parameters>
25             <!-- 최소 압축 크기 (단위: byte)-->
26             <CompressThreshold>1024</CompressThreshold>
27             <!-- 정보의 유효기간 (단위: Minute)-->
28             <Expiration>1440</Expiration>
29             <!-- MongoDB 작업을 위한 Repository -->
30             <Repository>${MongoRepository.PageState}</Repository>
31         </parameters>
32     </component>
33
34     <component id="MongoRepository.PageState"
35         service="RCL.Mongo.IMongoRepository, RCL.Mongo"
36         type="RCL.Mongo.MongoRepositoryImpl, RCL.Mongo"
37         lifestyle="Thread">
38         <parameters>
39             <!-- MongoDB connection string | app.config의 connection string name -->
40             <connStr>mongodb://localhost/RCLState?strict=false&pooling=true&poolSize=10</connStr>
41         </parameters>
42     </component>
43 </components>
44 </configuration>
```