we can transfer viewstate of page to another page.  
Here is the code:  
Create two pages one.aspx & two.aspx  
in one.aspx page  
protected void Page\_Load(object sender, EventArgs e)  
{  
ViewState["page1"] = "page1 viewstate";  
Server.Transfer("two.aspx");  
}  
  
public StateBag ReturnViewState()  
{  
return ViewState;  
}  
As you can see, I have set a ViewState variable in Page Load and transfer the user to two.aspx page using the Server.transfer() method. This page also contains a method ReturnViewState() which actually returns the ViewState of this page to the calling function. The return type of the method is StateBag class.  
  
StateBag class: This class is the primary storage mechanism for all HTML and Web server controls. It stores attribute/value pairs as strings associated with the control. It tracks changes to these attributes only after the OnInit method is executed for a page request, and saves the changes to the page’s or control’s viewstate.  
  
IN two.aspx page  
protected void Page\_Load(object sender, EventArgs e)  
{  
if (PreviousPage != null)  
{  
if (PreviourPageViewstate() != null)  
{  
Label1.Text =Convert.ToString(PreviourPageViewstate()["page1"]);  
}  
}  
}  
  
private StateBag PreviourPageViewstate()  
{  
  
StateBag returnValue = null;  
if (PreviousPage != null)  
{  
object prePage = (object)PreviousPage;  
MethodInfo objMethod = prePage.GetType().GetMethod("ReturnViewState");  
return (StateBag)objMethod.Invoke(prePage, null);  
}  
return returnValue;  
}  
Whenever we use Server.transfer or Cross Page Posting, We can get the previous page object via PreviousPage property. Using Previous Page, we can find the controls of the previous page. For example, one can access Label control placed in ViewStateContainer Page in current Page.  
  
Looking at the code, I have created a PreviousPageViewState property in this page, which returns the previous page’s ViewState. It first checks whether PreviousPage is null or not, if it’s not null, then it creates an object of the previous page.  
  
Now using Reflection, we can invoke the method of the previous class. Using MethodInfo class, I have invoked the ReturnViewState() method of ViewStateContainer Page.  
  
In Page\_Load event, I am able to access the ViewState variable of ViewStateContainer Page. You can access all the viewstate variables set in ViewStateContainer Page.  
  
Run the application & see the effect