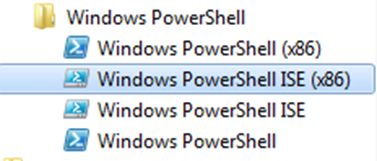
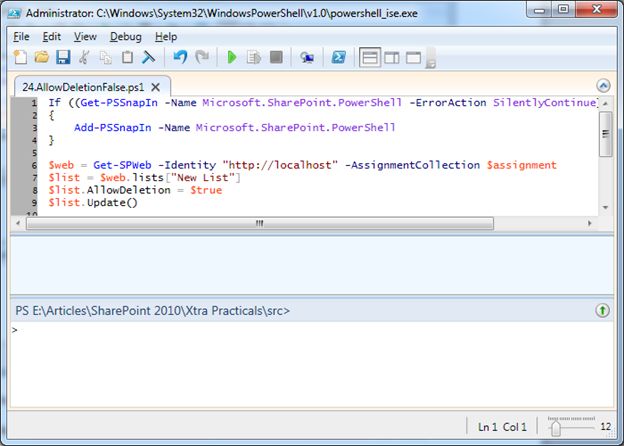
**PowerShell Approach**  
  
Now we can try to reverse the AllowDeletion property using PowerShell script. Please note that for these types of scenarios on client locations PowerShell would be more handy. (Visual Studio might not be installed in the customer premises.)  
  
You can start the PowerShell Editor aka ISE (Integrated Script Editor) from Start > Programs > Accessories > Windows Power Shell menu group.

  
  
The editor is shown below. Save the default file as AllowDeletionFalse.ps1. (ps1 is the extension for PowerShell scripts.)  
  
  
  
The following is the same as shown above in the .ps1 file:

If ((Get-PSSnapIn -Name Microsoft.SharePoint.PowerShell -ErrorAction SilentlyContinue) -eq $null )

{

Add-PSSnapIn -Name Microsoft.SharePoint.PowerShell

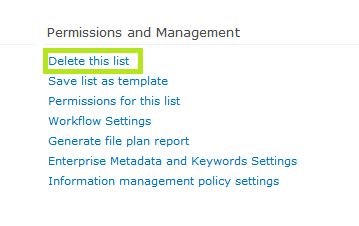
}

$web = Get-SPWeb -Identity "http://localhost" -AssignmentCollection $assignment

$list = $web.lists["New List"]

$list.AllowDeletion = $true

$list.Update()

**Code Explained**  
  
The first If block make sure that the Microsoft.SharePoint.PowerShell snap-in is loaded in to memory. (It is similar to add reference in Visual Studio.)  
  
The Get-SPWeb cmdlet retrieves the reference to specified web application. The web.lists returns the reference to our New List. After setting the AllowDeletion to true the list is updated. You can use the Run command of the PowerShell editor to execute the code.  
  
DltShr7.jpg  
  
Once the code has executed successfully our New List should have the Delete this list link visible in the List Settings. You can refresh the List Settings page to ensure that it is restored.  
  


**References**  
  
<http://tinyurl.com/sp2010-list-allow-del>   
  
**Summary**  
  
In this article we have seen how to protect a list by disabling the Delete list option using Visual Studio and PowerShell. The source code is contained in the attachment.