

SharePoint 2010 的安装手册和指南已经很多了,但是查看了几份却没有找到在 Windows 7 下安装 SharePoint Server 2010 的,正好需要安装这样的环境,顺便写下来分享给大家。

## 1 安装环境

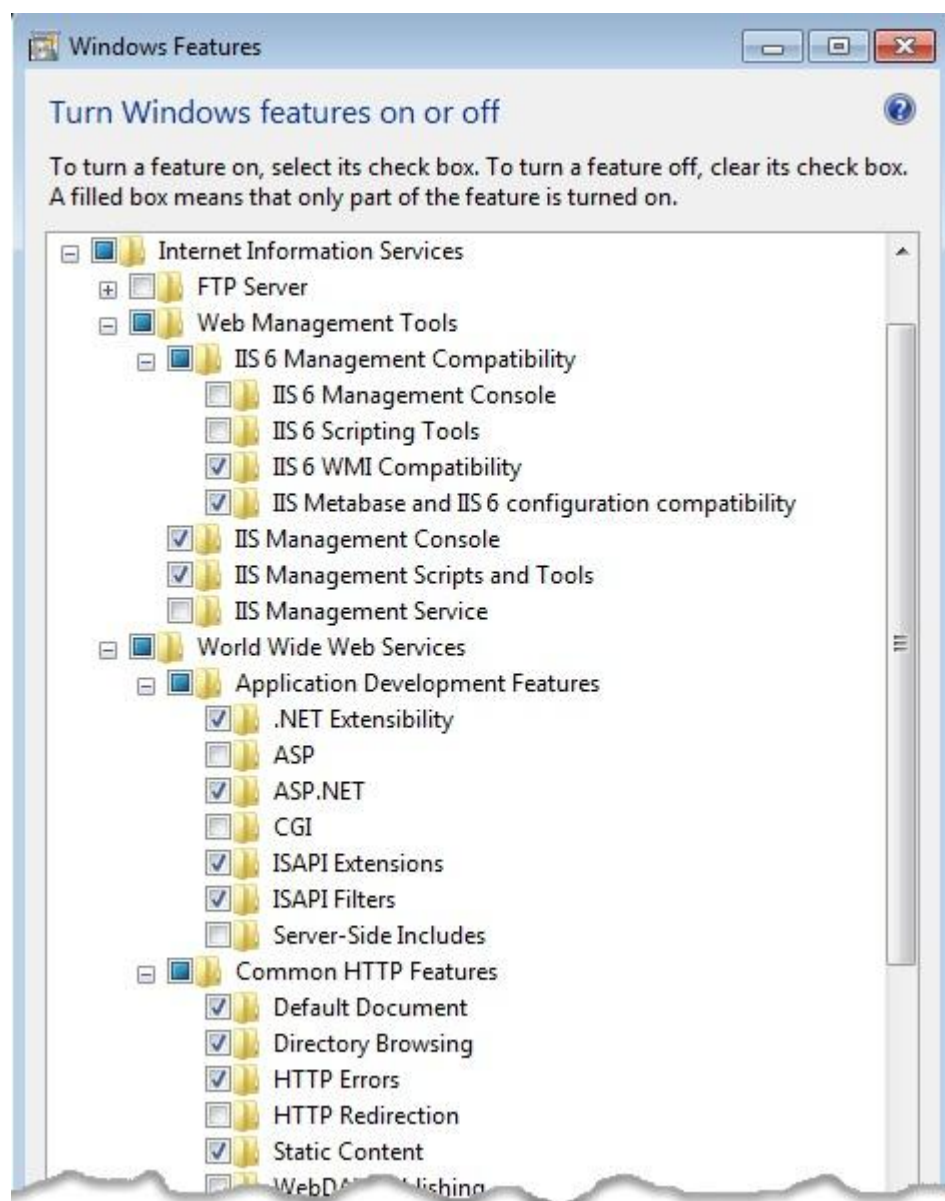
具体的安装环境如下所示:

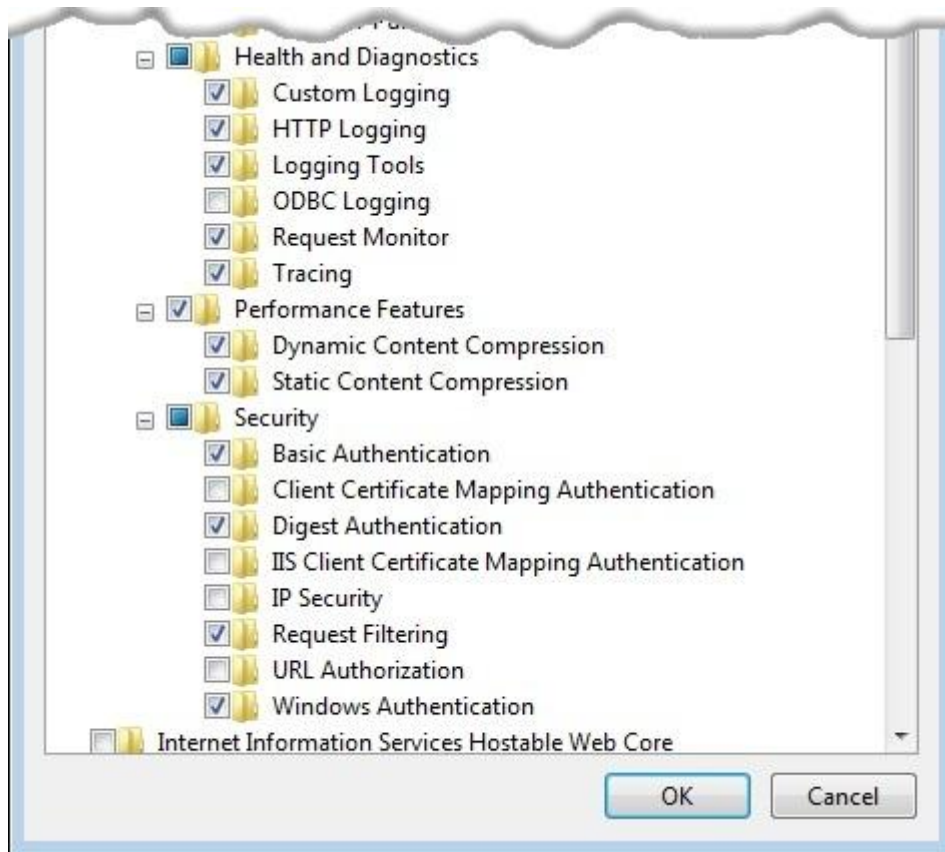
操作系统: Windows 7 enterprise X64 with service pack 1, 64 位英文版, 因为 SharePoint 2010 只支持 64 为系统, 这和 MOSS 2007 是不同的。

数据库: MS SQL Server 2008 R2 英文版

SharePoint Server 2010: MSDN 上下载的英文版

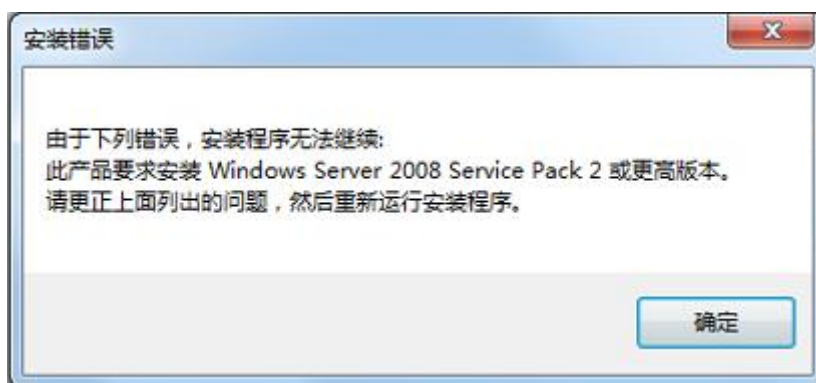
IIS 配置:





## 2 安装必备

SharePoint 2010 支持 Windows Server 2008 Service Pack 2 和 Windows Server 2008 R2 下 X64 系统的安装，不直接支持 Windows 7 下的安装，若我们在虚拟光驱直接点击 Setup.Exe（SharePoint Server 2010 的安装文件，SharePoint Foundation 2010 的安装文件为 SharePointFoundation.Exe）时，会提示一下的错误：



这对于使用 Windows 7 的而且又想玩 SharePoint 2010 的兄弟来说可不是好消息，不过幸运的是我们可以修改配置文件，手动安装必备组件的方式绕开微软为安装 SharePoint 2010 设置的这个门槛，下面我们就分别来描述。

## 修改 SharePoint Server 2010 配置

1. 创建一个文件夹作为存放安装文件的路径，如 c:\SharePoint Server 2010
2. 把下载的 ISO 文件使用虚拟光驱虚拟出来，打开虚拟光驱虚拟的文件，全选|复制到上一步创建的文件夹中
3. 找到\\SharePointServer2010\Files,修改此文件夹下所有的 config.xml 文件为:

```
<Configuration>
  <Package Id="sts">
    <Setting Id="LAUNCHEDFROMSETUPSTS" Value="Yes"/>
  </Package>

  <Package Id="spswfe">
    <Setting Id="SETUPCALLED" Value="1"/>
  </Package>

  <Logging Type="verbose" Path="%temp%" Template="SharePoint Server Setup(*).log"/>
  <!--<PIDKEY Value="Enter Product Key Here" />-->
  <Setting Id="SERVERROLE" Value="SINGLESERVER"/>
  <Setting Id="USINGUIINSTALLMODE" Value="1"/>
  <Setting Id="SETUPTYPE" Value="CLEAN_INSTALL"/>
  <Setting Id="SETUP_REBOOT" Value="Never"/>
  <Setting Id="AllowWindowsClientInstall" Value="True"/>
</Configuration>
```

注意:

Files 下面有多个文件夹: Setup、SerupFarm、SetupFarmSlient、SetupFarmUpgrade、SetupSlient、SetupSingleUpgrade, 这些文件夹下的 config.xml 文件都替换成上面的 xml

## 安装必备组件

必备组件的下载位置参考:

### SharePoint Server

以下这些组件适用于 Windows Server 2008 及 Windows Server 2008 R2。

KB971831 (目前该 KB 还没有对 2008 R2 进行更新, 这个 KB 为 WCF 添加一个新模式 "AllowInsecureTransport", 如果没有这个模式则 User Profile SA、Metadata Service SA 不能正常工作)

<http://support.microsoft.com/kb/971831>

<http://go.microsoft.com/fwlink/?LinkID=160770>

Microsoft SQL Server 2008 Native Client:

<http://download.microsoft.com/download/3/5/5/35522a0d-9743-4b8c-a5b3-f10529178b8a/sqlncli.msi>

Microsoft "Geneva" Framework Runtime

<http://download.microsoft.com/download/F/3/D/F3D66A7E-C974-4A60-B7A5-382A61EB7BC6/MicrosoftGenevaFramework.amd64.msi>

Microsoft Sync Framework Runtime v1.0 (x64)

<http://download.microsoft.com/download/C/9/F/C9F6B386-824B-4F9E-BD5D-F95BB254EC61/Redist/amd64/Microsoft%20Sync%20Framework/Synchronization.msi>

Microsoft Chart Controls for Microsoft .NET Framework 3.5

<http://download.microsoft.com/download/c/c/4/cc4dcac6-ea60-4868-a8e0-62a8510aa747/MSChart.exe>

注意:

MSChart.exe 要下载 Microsoft Chart Controls for Microsoft .NET Framework 3.5 下的, 参考文件里下载的 exe 安装可能会有问题。

Microsoft SQL Server 2008 Analysis Services ADOMD.NET

[http://download.microsoft.com/download/A/D/0/AD021EF1-9CBC-4D11-AB51-6A65019D4706/SQLSERVER2008\\_ASADOMD10.msi](http://download.microsoft.com/download/A/D/0/AD021EF1-9CBC-4D11-AB51-6A65019D4706/SQLSERVER2008_ASADOMD10.msi)

Filter Pack 2.0 在安装文件中自带, 通常无需另行下载。

在 Windows Server 2008 上安装准备时, 需要以下这些额外的组件:

.Net Framework 3.5 SP1

[.NET Framework 3.5 Service Pack 1 \(Full Package\)](#) [KB959209](#) [KB967190](#)

PowerShell V2 CTP3 (Win 7, 2008 R2 中自带了 RTM。在 2008 上最好也安装 RTM, 但目前还没有发布)

[http://download.microsoft.com/download/D/0/E/D0E6D2C1-2593-4017-B26D-7375BC9263D5/PowerShell\\_Setup\\_amd64.msi](http://download.microsoft.com/download/D/0/E/D0E6D2C1-2593-4017-B26D-7375BC9263D5/PowerShell_Setup_amd64.msi)

SQL Server 2005 补丁

SQL Server 2005 SP3

<http://www.microsoft.com/downloads/details.aspx?familyid=AE7387C3-348C-4FAA-8AE5-949FDFBE59C4&displaylang=en>

CU3 for SQL Server 2005 SP3

<http://support.microsoft.com/kb/967909>

SQL Server 2008 补丁

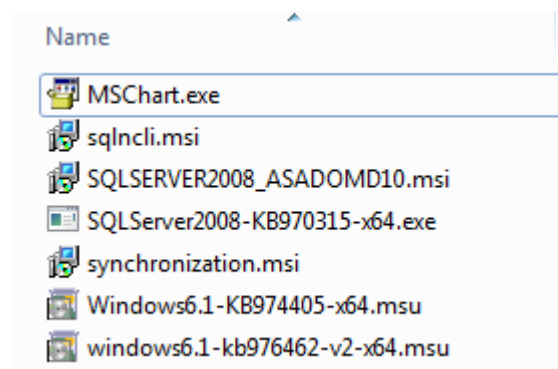
SQL Server 2008 SP1

<http://www.microsoft.com/downloads/details.aspx?familyid=66AB3DBB-BF3E-4F46-9559-CCC6A4F9DC19&displaylang=en>

CU2 for SQL Server 2008 SP1

[Cumulative update package 2 for SQL Server 2008 Service Pack 1](#)

最终我下载的必备组件如下图所示：



一一安装上这些必备组件就可以进行 SharePoint Server 2010 的正式安装了。

## 3 安装并配置 SharePoint Server 2010

### 3.1 运行 PowerShell 安装脚本

找到安装 SharePoint 2010 的 PowerShell 脚本文件，右击选择“Run with PowerShell”，出现运行脚本的窗口，等待 1-2 分钟以后自动关闭该窗口。运行的 PowerShell 脚本如下：

```
#####  
#####
```

```

#This script installs SharePoint 2010 on Windows 7, Do not run this script to install SharePoint
2010
#on Windows Server 2008 or Windows 2008 R2 or any other Operating Systems.
#
# This script will prompt to enter Setup Dir path and SharePoint Server 2010 Product Key to be
used when
# Installing
#
# Step 1 Add/Remove windows Components
# Step 2 Extract the OfficeServer.exe file
# Step 3 Downloads Pre-Reqs from Microsoft Downloads site
# Step 4 Install Pre-Reqs
# Step 5 Creates a Config.xml file for Windows Client Install (SingleServer mode)
# Step 6 Install SharePoint 2010
#*****
*****
param([string]$SetupDir = $(Read-Host -Prompt "Please Enter Directory Path where SharePoint
2010 Setup.exe is")
,[string]$PidKey = $(Read-Host -Prompt "Enter SharePoint Server 2010 Product Key"))

Import-Module BitsTransfer

$DownloadUrls = (

"http://download.microsoft.com/download/C/9/F/C9F6B386-824B-4F9E-BD5D-F95BB254EC61/R
edist/amd64/Microsoft%20Sync%20Framework/Synchronization.msi", #
http://go.microsoft.com/fwlink/?LinkID=141237&clcid=0x409 - Microsoft Sync Framework
Runtime v1.0 (x64)

"http://download.microsoft.com/download/3/5/5/35522a0d-9743-4b8c-a5b3-f10529178b8a/sql
ncli.msi", # "http://go.microsoft.com/fwlink/?LinkId=123718&clcid=0x409" - SQL Server 2008
Native Client

"http://download.microsoft.com/download/D/7/2/D72FD747-69B6-40B7-875B-C2B40A6B2BDD/
Windows6.1-KB974405-x64.msu", # "http://go.microsoft.com/fwlink/?LinkID=166363" -
Windows Identity Framework (Win2008 R2)

"http://download.microsoft.com/download/1/7/1/171CCDD6-420D-4635-867E-6799E99AB93F/A
DONETDataServices_v15_CTP2_RuntimeOnly.exe", #
"http://go.microsoft.com/fwlink/?LinkId=158354" - ADO.NET Data Services v1.5 CTP2 (Win2008
SP2)

"http://download.microsoft.com/download/c/c/4/cc4dcac6-ea60-4868-a8e0-62a8510aa747/MS

```

```
Chart.exe", # "http://go.microsoft.com/fwlink/?LinkID=141512" - Microsoft Chart Controls for the  
Microsoft .NET Framework 3.5
```

```
"http://download.microsoft.com/download/b/9/a/b9a78047-d269-4a25-88a1-9c4321d90677/S  
QLSERVER2008_ASADOMD10.msi" #
```

```
"http://go.microsoft.com/fwlink/?LinkId=130651&clcid=0x409" - Microsoft SQL Server 2008  
Analysis Services ADOMD.NET  
)
```

```
function AddRemoveWindowsComponents()  
{  
    Write-Host "Enabling required windows components"  
  
    $ReturnCode = 0  
    $WindowsFeatures = (  
        "IIS-WebServerRole",  
        "IIS-WebServer",  
        "IIS-CommonHttpFeatures",  
        "IIS-StaticContent",  
        "IIS-DefaultDocument",  
        "IIS-DirectoryBrowsing",  
        "IIS-HttpErrors",  
        "IIS-ApplicationDevelopment",  
        "IIS-ASPNET",  
        "IIS-NetFxExtensibility",  
        "IIS-ISAPIExtensions",  
        "IIS-ISAPIFilter",  
        "IIS-HealthAndDiagnostics",  
        "IIS-HttpLogging",  
        "IIS-LoggingLibraries",  
        "IIS-RequestMonitor",  
        "IIS-HttpTracing",  
        "IIS-CustomLogging",  
        "IIS-ManagementScriptingTools",  
        "IIS-Security",  
        "IIS-BasicAuthentication",  
        "IIS-WindowsAuthentication",  
        "IIS-DigestAuthentication",  
        "IIS-RequestFiltering",  
        "IIS-Performance",  
        "IIS-HttpCompressionStatic",  
        "IIS-HttpCompressionDynamic",  
        "IIS-WebServerManagementTools",
```

```

        "IIS-ManagementConsole",
        "IIS-IIS6ManagementCompatibility",
        "IIS-Metabase",
        "IIS-WMICompatibility",
        "WAS-WindowsActivationService",
        "WAS-ProcessModel",
        "WAS-NetFxEnvironment",
        "WAS-ConfigurationAPI",
        "WCF-HTTP-Activation",
        "WCF-NonHTTP-Activation"
    )
}

Try
{
    $StartInfo = New-Object -TypeName System.Diagnostics.ProcessStartInfo
    $StartInfo.FileName = "$Env:windir\System32\pkgmgr.exe"
    $StartInfo.WindowStyle = [System.Diagnostics.ProcessWindowStyle]::Minimized
    $WindowsfeaturesString = [string]::join(";", $WindowsFeatures)
    $pArgs = "/iu:$WindowsfeaturesString /quiet"
    Write-Host $pArgs
    $StartInfo.Arguments = $pArgs
    $StartTime = [System.DateTime]::Now
    $Step = 5
    $Val = 0
    $InstallProcess = [System.Diagnostics.Process]::Start($StartInfo)

    While( -not($InstallProcess.HasExited))
    {
        $CurrentRuntime = [System.DateTime]::Now - $StartTime
        Write-Progress -Activity "Add/Remove Windows Components..."
        -PercentComplete ($Val+=$Step) -Status ("Add/Remove windows components has been running
for {0}:{1}:{2}" -f $CurrentRuntime.Hours.ToString("00"),
$CurrentRuntime.Minutes.ToString("00"), $CurrentRuntime.Seconds.ToString("00"))

        if ($Val -ge 100) { $Val = 0}
        Start-Sleep -Seconds 1
    }
    Write-Host "Done enabling all required windows components"
}
Catch
{
    $ReturnCode = -1
    Write-Warning "Error Add/Remove Windows Components"
    Write-Error $_
    Break;
}

```



```

    }

    return $ReturnCode
}

function DownloadPreRequisites()
{
    $ReturnCode = 0
    Write-Host "Downloading Pre-Reqs required for SharePoint 2010 Windows Client Install"

    #download sharepoint 2010 prerequisites
    $PreReqInstallerDir = "$SetupDir\PrerequisiteInstallerFiles"

    ForEach ($DownloadUrl in $DownloadUrls)
    {
        ## Get the file name based on the portion of the URL after the last slash
        $FileName = $DownloadUrl.Split('/')[-1]
        Try
        {
            ## Check if destination file already exists
            If (!(Test-Path "$PreReqInstallerDir\$FileName"))
            {
                ## Begin download
                Start-BitsTransfer -Source $DownloadUrl -Destination
$PreReqInstallerDir\$fileName -DisplayName "Downloading ``$FileName`` to $PreReqInstallerDir"
-Priority High -Description "From $DownloadUrl..." -ErrorVariable err
                If ($err) {Throw ""}
            }
            Else
            {
                Write-Host " - File $FileName already exists, skipping..."
            }
        }
        Catch
        {
            $ReturnCode = -1
            Write-Warning " - An error occurred downloading ``$FileName``"
            Write-Error $_
            break
        }
    }

    Write-Host "Done downloading Pre-Reqs required for SharePoint 2010 windows client
install"

```

```

return $ReturnCode
}

function InstallPreReqs()
{
    Write-Host "Installing Pre-Reqs required for SharePoint 2010 windows client install"
    $PreReqInstallerDir = "$SetupDir\PrerequisiteInstallerFiles"

    #install filter pak
    $Filterpak = "$PreReqInstallerDir\FilterPack\FilterPack.msi"
    If (Test-Path -Path $Filterpak)
    {
        Try
        {
            Write-Host "Installing FilterPack"
            $StartInfo = New-Object -TypeName System.Diagnostics.ProcessStartInfo
            $StartInfo.FileName = $filterpak
            $StartInfo.WindowStyle = [System.Diagnostics.ProcessWindowStyle]::Minimized
            $InstallProcess = [System.Diagnostics.Process]::Start($StartInfo);

            While(-not($InstallProcess.HasExited))
            {
                Start-Sleep -Seconds 1
            }
            Write-Host "FilterPack installed successfully"
        }
        Catch
        {
            $ReturnCode = -1
            Write-Warning " - An error occurred while installing FilterPak"
            Write-Error $_
            break
        }
    }
    else
    {
        Write-Host "FilterPack install bits not found, exiting..."
        break;
    }

    #Install rest of the Pre-Reqs that was downloaded for Windows Client Install
    ForEach ($DownloadUrl in $DownloadUrls)
    {
        ## Get the file name based on the portion of the URL after the last slash

```

```

$FileName = $DownloadUrl.Split('/')[-1]
Try
{
    ## Check if destination file already exists
    If (Test-Path "$PreReqInstallerDir\$FileName")
    {
        $StartInfo = New-Object -TypeName System.Diagnostics.ProcessStartInfo
        $StartInfo.FileName = "$PreReqInstallerDir\$FileName"
        $StartInfo.WindowStyle = [System.Diagnostics.ProcessWindowStyle]::Minimized
        $InstallProcess = [System.Diagnostics.Process]::Start($StartInfo);

        While(-not($InstallProcess.HasExited))
        {
            Start-Sleep -Seconds 1
        }
        Write-Host "$FileName installed successfully"
    }
}
Catch
{
    $ReturnCode = -1
    Write-Warning " - An error occurred while installing '$FileName'"
    Write-Error $_
    break
}
}
return $ReturnCode
}

function ExtractSP2010SetupExe()
{
    $ReturnCode = 0

    Write-Host "Extracting SharePoint 2010 bits for Windows Client Install"

    Try
    {
        if( -not(Test-Path $SetupDir -Verbose))
        {
            Write-Host "Invalid Setup Directory, Exiting..."
            break;
        }
    }
    $StartTime = [System.DateTime]::Now

```

```

$StartInfo = New-Object -TypeName System.Diagnostics.ProcessStartInfo
$StartInfo.FileName = Join-Path $SetupDir "OfficeServer.exe"
$StartInfo.Arguments = "/quiet /extract:$SetupDir"
$StartInfo.WindowStyle = [System.Diagnostics.ProcessWindowStyle]::Minimized
$ExtractProcess = [System.Diagnostics.Process]::Start($StartInfo);

$Step = 5
$Val = 0
while(-not($ExtractProcess.HasExited))
{
    $CurrentRuntime = [System.DateTime]::Now - $StartTime
    Write-Progress -Activity "Extracting files..." -PercentComplete ($Val+=$Step)
-Status ("Extract process has been running for {0}:{1}:{2}" -f
$CurrentRuntime.Hours.ToString("00"), $CurrentRuntime.Minutes.ToString("00"),
$CurrentRuntime.Seconds.ToString("00"))

    if ($Val -ge 100) { $Val = 0}
    Start-Sleep -Seconds 1
}
}
Catch
{
    $ReturnCode = -1
    Write-Warning "Error Extracting Setup Files"
    Write-Error $_
    Break;
}
Write-Host "Done Extracting SharePoint 2010 bits for Windows Client Install"
}

function InstallSharePoint2010()
{
    $rc = 0
    #Add Remove Windows Components
    $rc = AddRemoveWindowsComponents

    #Extract Setup.exe
    if($rc -ne -1)
    {
        $rc = ExtractSP2010SetupExe
    }

    #Download Pre-Reqs for Windows Client Install
    if($rc -ne -1)

```

```

{
    $Src = DownloadPreRequisites
}

#Install the Pre-Reqs
if($Src -ne -1)
{
    $Src = InstallPreReqs
}

Try
{
    #SetUp Config File for Installing SharePoint 2010 in Silent Mode on Windows Client
    [System.Xml.XmlDocument]$XmlDoc =
    @"
    <Configuration>
        <Logging Type="verbose" Path="%temp%" Template="SharePoint Server
Setup(*).log"/>
        <Display Level="none" CompletionNotice="no" AcceptEula="no"
SuppressModal="yes" NoCancel="yes"/>
        <PIDKEY Value="Enter PID Key Here" />
        <Setting Id="SERVERROLE" Value="SINGLESERVER"/>
        <Setting Id="USINGUIINSTALLMODE" Value="0"/>
        <Setting Id="SETUP_REBOOT" Value="Never" />
        <Setting Id="SETUPTYPE" Value="CLEAN_INSTALL"/>
    </Configuration>
"@

    #Set the SharePoint 2010 Product Key
    Write-Host "Setting PIDKey in Config File"
    $XmlDoc.Configuration.PIDKEY.Value = $PidKey

    #update the config file to allow windows client install. <Setting
    Id="AllowWindowsClientInstall" Value="True"/>
    Write-Host "Setting AllowWindowsClientInstall in Config File"
    $ClientOSNode = $XmlDoc.CreateNode([System.Xml.XmlNodeType]::Element,
    "Setting", $null);
    $ClientOSNode.SetAttribute("Id", "AllowWindowsClientInstall")
    $ClientOSNode.SetAttribute("Value", "True")
    $XmlDoc.Configuration.AppendChild($ClientOSNode) | Out-Null

    $ConfigXmlDir = "$SetupDir\Files\SetupSilent\"
    $ConfigXmlName = ("config_{0}.xml" -f (Get-Date -Format
'yyyy_MMM_dd_HH_mm_ss'))

```

```

$ConfigXmlPath = Join-Path $ConfigXmlDir $ConfigXmlName

Write-Host "Saving Config.xml file to $ConfigXmlPath"
$xmlDoc.Save($ConfigXmlPath)

#Install SharePoint 2010
Write-Host "Installing SharePoint 2010"
$SetupArgs = ("/config {0}" -f $ConfigXmlPath)
Write-Host "Setup Arguments :$SetupArgs"
$SetupExePath = Join-Path $SetupDir "Setup.exe"
Write-Host "Running Command : $SetupExePath"

$StartTime = [System.DateTime]::Now
$Step = 5
$Val = 0

$SetupExe = [System.Diagnostics.Process]::Start($SetupExePath, $SetupArgs)
While(-not($SetupExe.HasExited))
{
    $CurrentRuntime = [System.DateTime]::Now - $StartTime
    Write-Progress -Activity "Installing SharePoint..." -PercentComplete ($Val+=$Step)
-Status ("Setup has been running for {0}:{1}:{2}" -f $CurrentRuntime.Hours.ToString("00"),
$CurrentRuntime.Minutes.ToString("00"), $CurrentRuntime.Seconds.ToString("00"))

    if ($Val -ge 100) { $Val = 0}
    Start-Sleep -Seconds 5
}

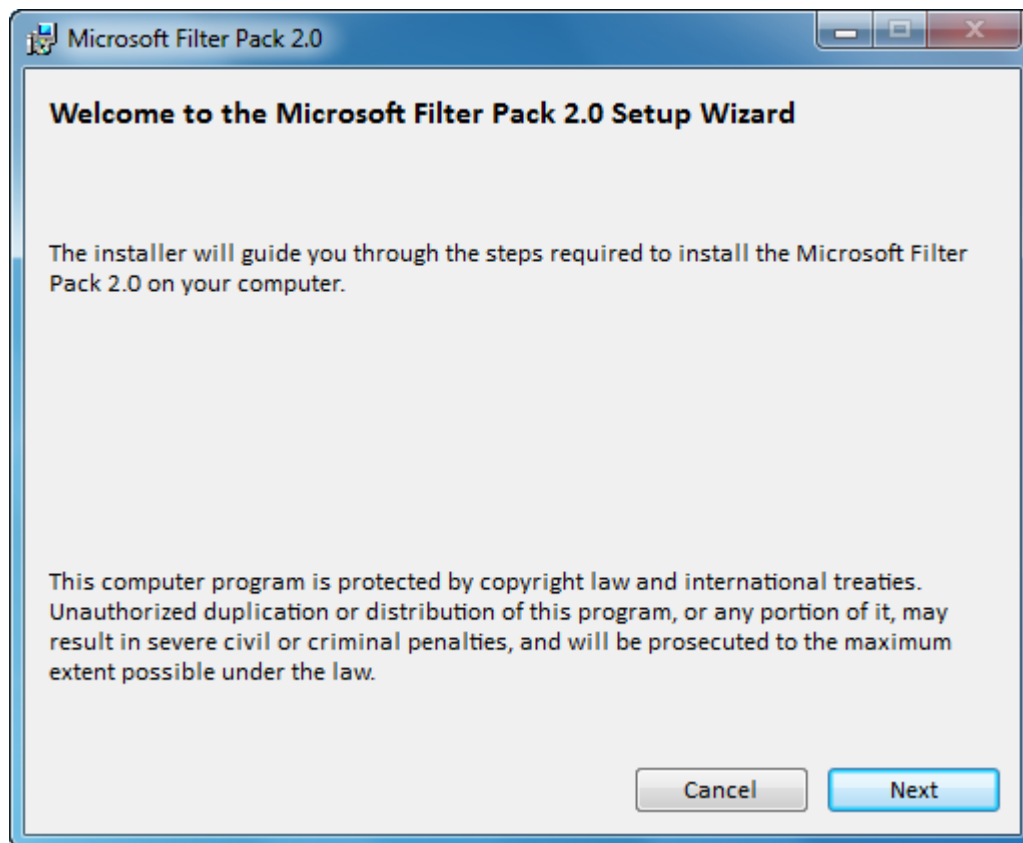
Write-Host "SharePoint 2010 Install Complete, Please remember to run the Farm
Configuration wizard"
}
Catch
{
    Write-Warning "Error - Installing SharePoint 2010"
    Write-Error $_
    Break;
}
}
InstallSharePoint2010

```

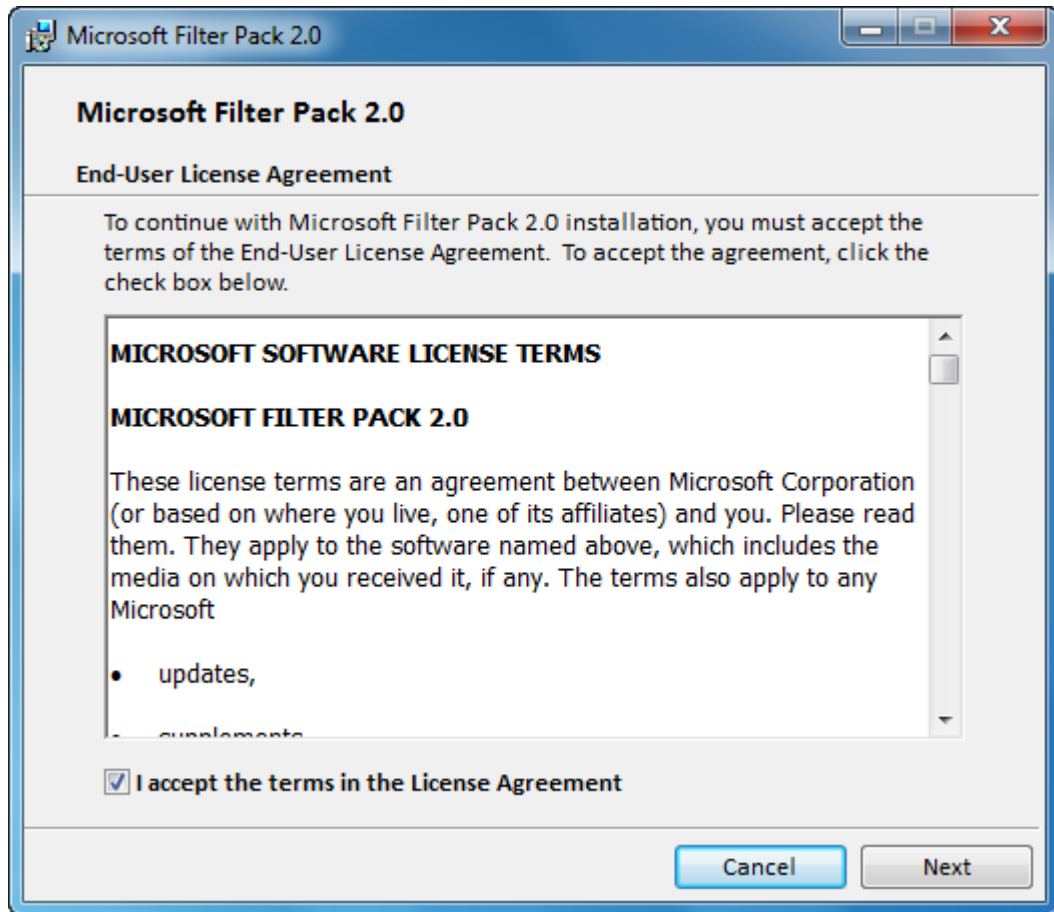
## 3.2 运行 FilterPack.MSI

在 [\\SharePointServer2010\PrerequisiteInstallerFiles\FilterPack](#) 目录下找到 FilterPack.MSI 文件，

双击并安装，界面如下：



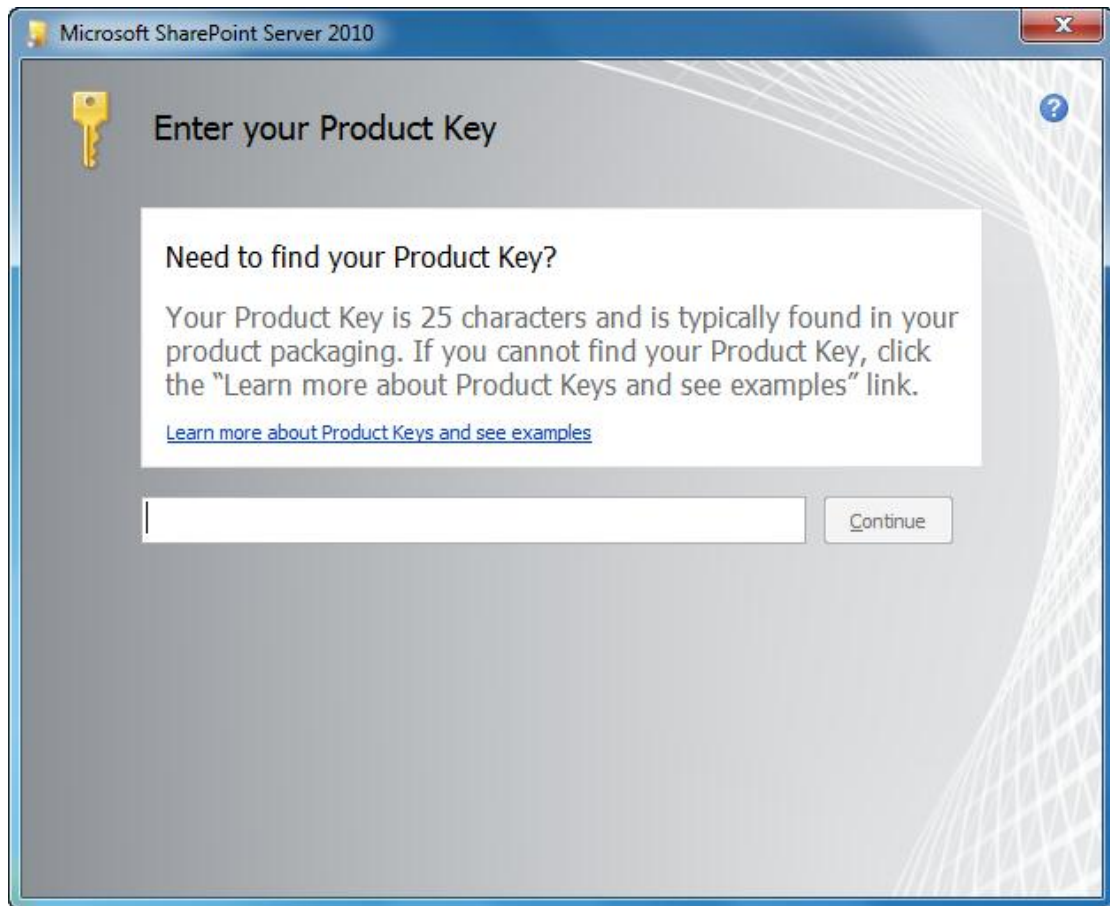
点击 Next，出现界面如下：



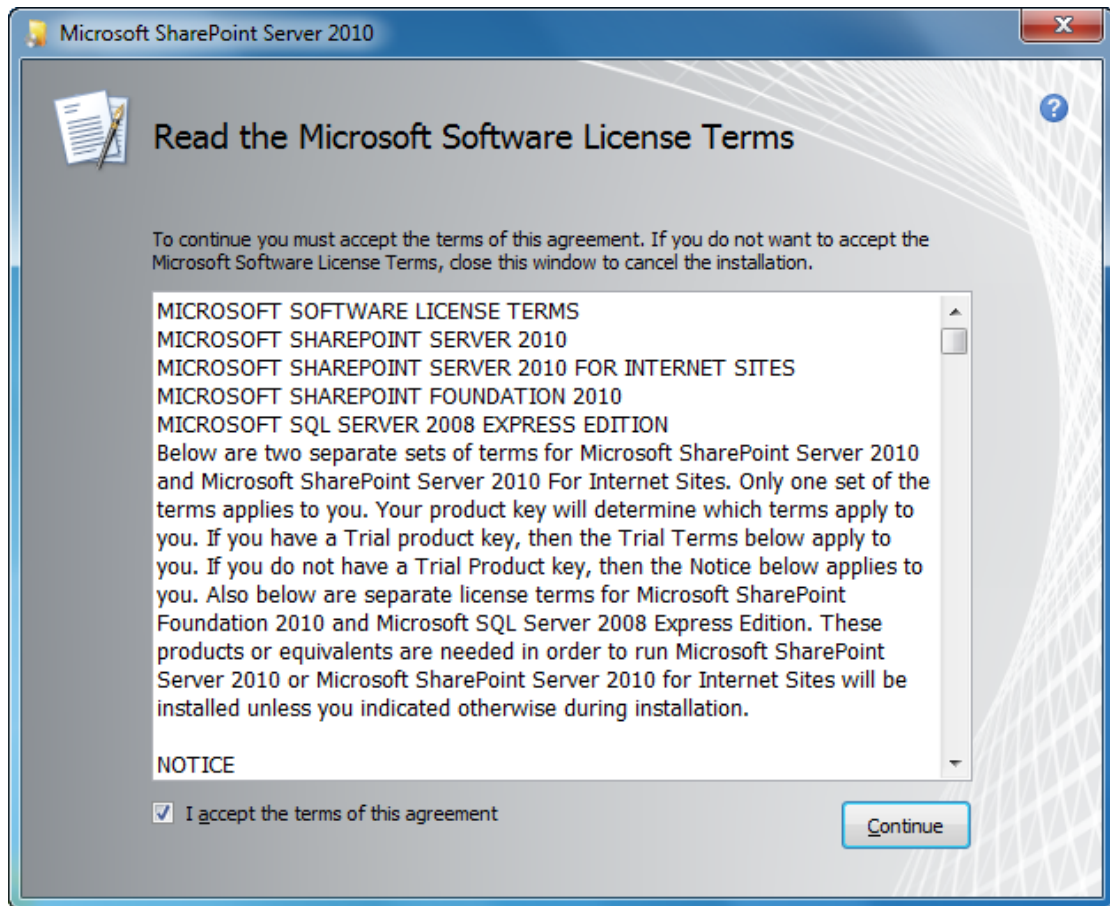
### 3.3 运行 Setup.Exe

运行 setup.exe，出现输入产品密钥的界面如下：

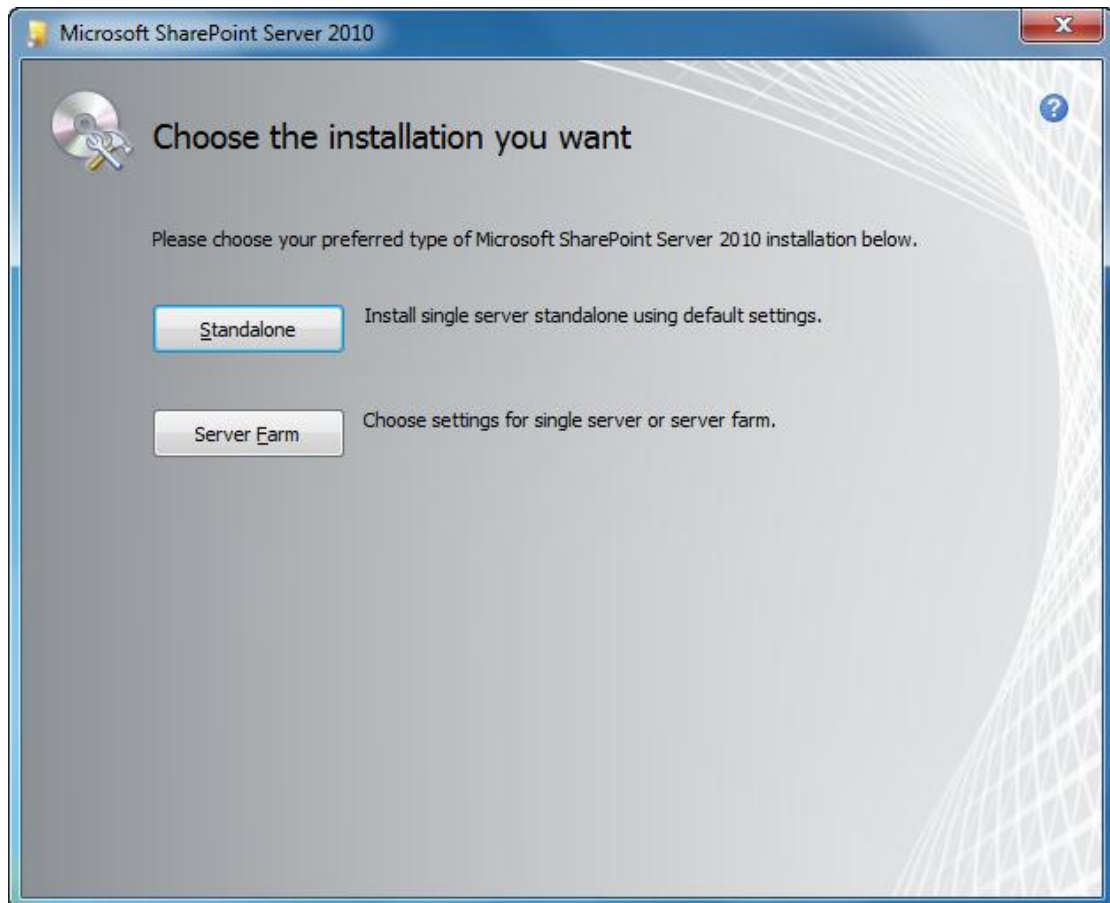




输入密钥，点击“continue”，界面如下：



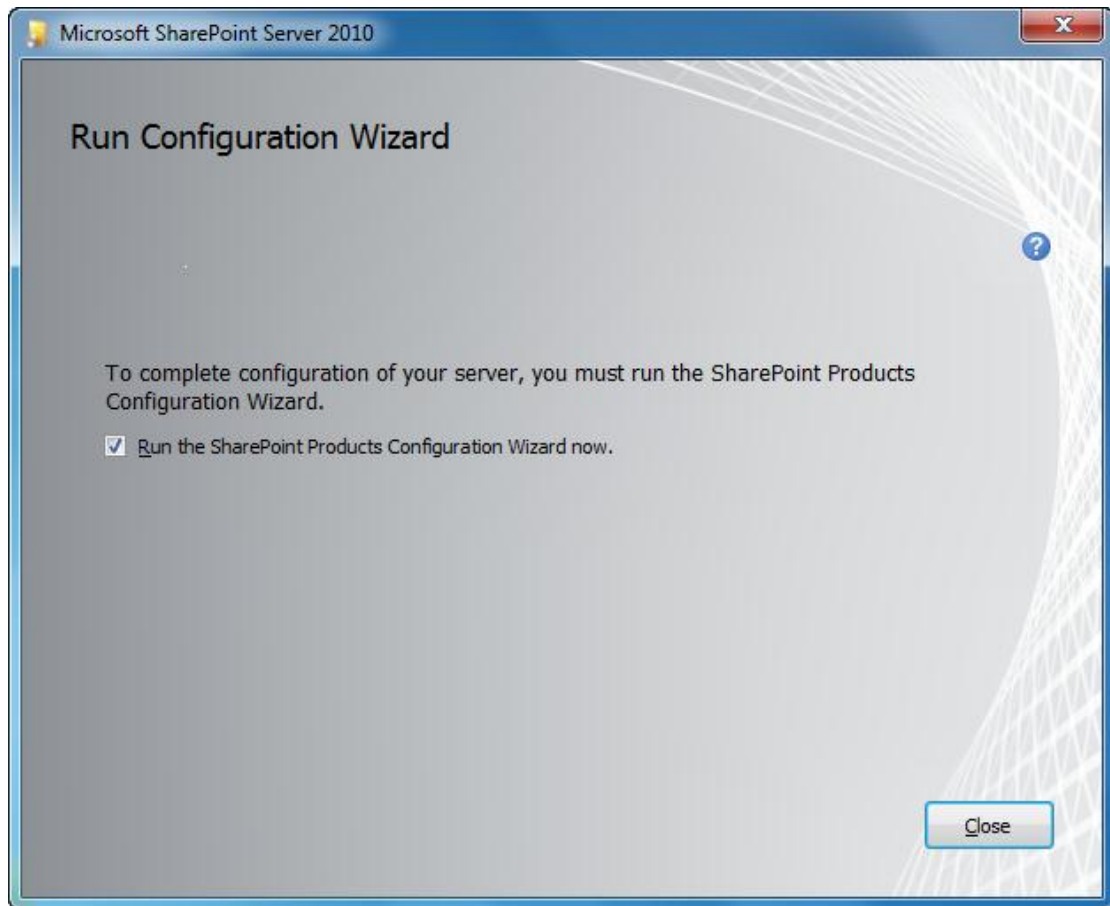
点击“Continue”，界面如下：



在这里选择了“Standalone，独立“安装，这样的情况下将会在我们的 SQL Server 服务器上创建一个名为”sharepoint“的实例，我们可以使用“.\sharepoint”在 SQL Server Management Studio 中打开。

当然，默认情况下我们是配置不了数据库的，也可以配置使用默认的实例来创建 SharePoint 的内容数据库，已有相关方案，本文就不讨论了。

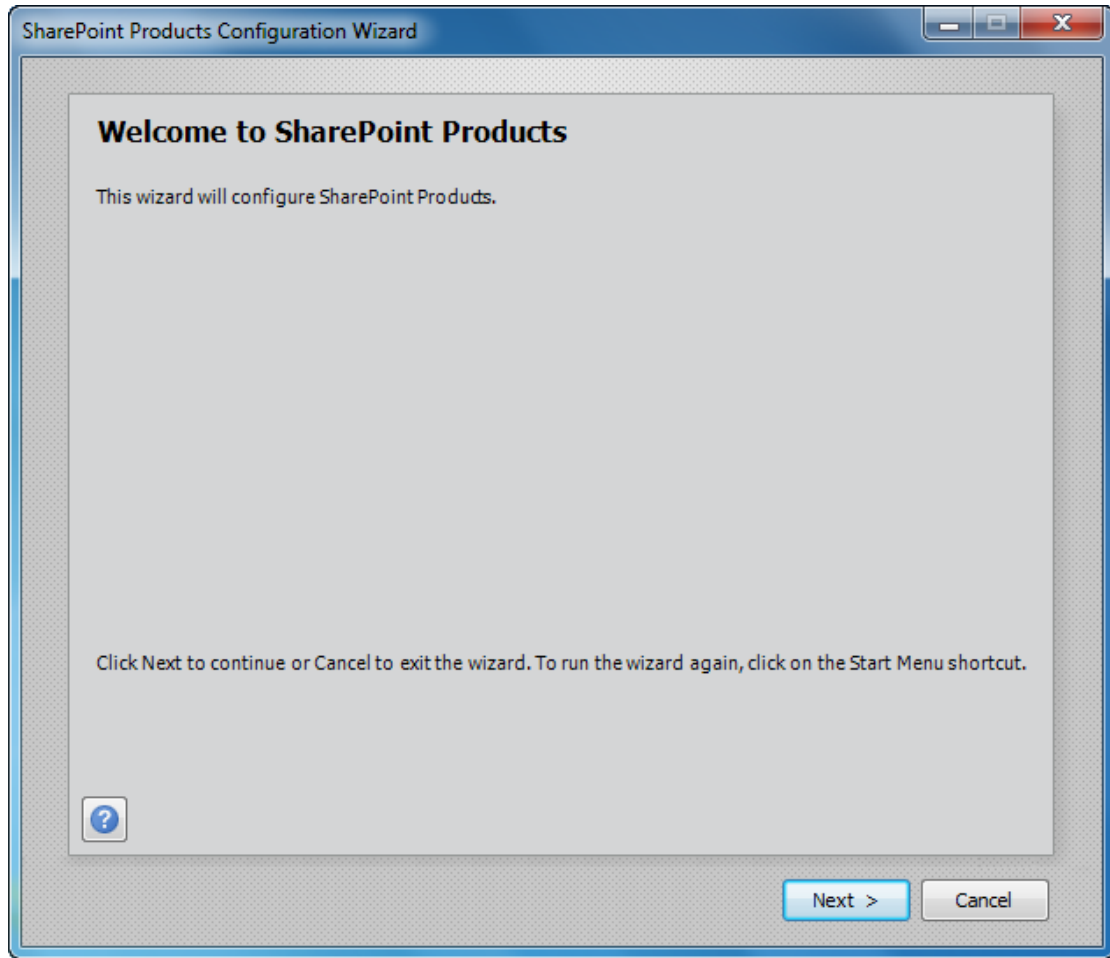
安装完成以后，弹出的窗口如下图所示：



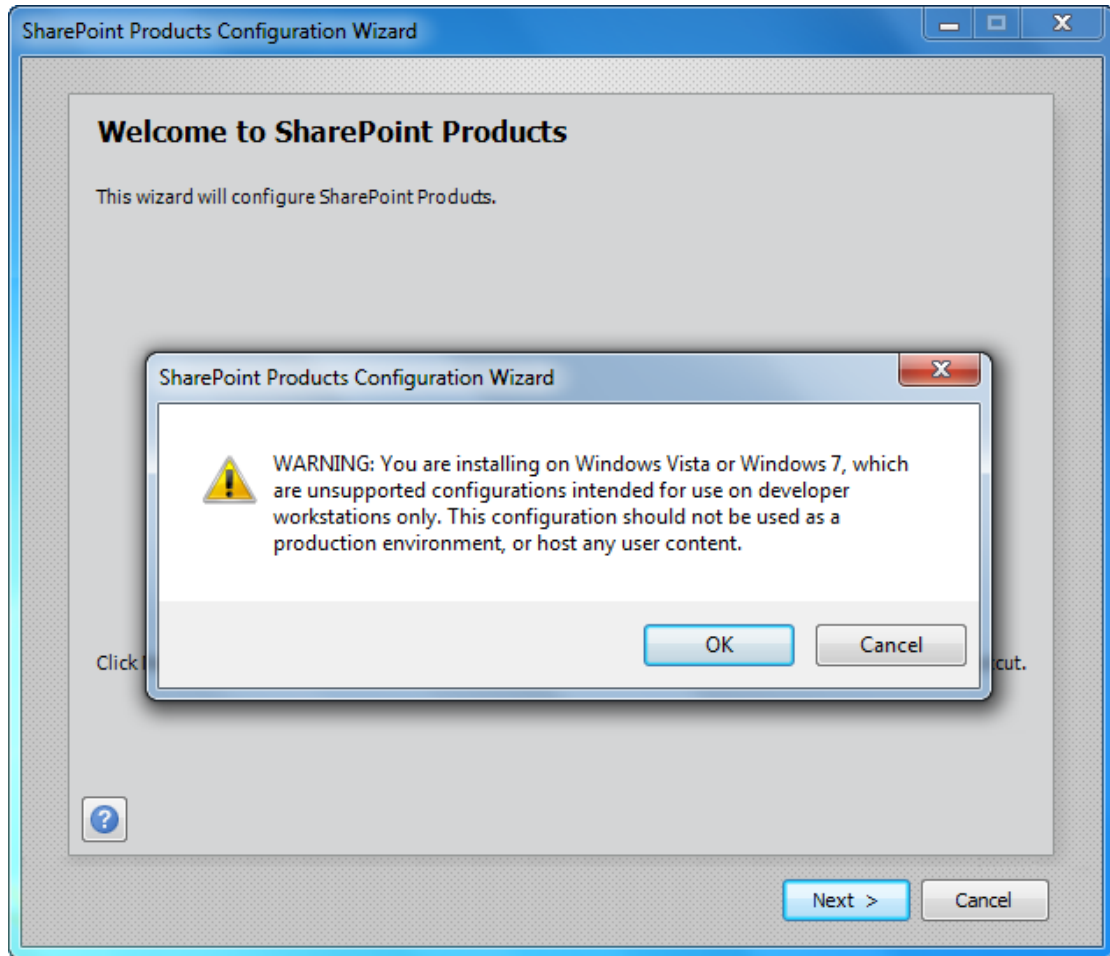
点击“Close”，开始配置 SharePoint Server 2010。

### 3.4 配置 SharePoint Server 2010

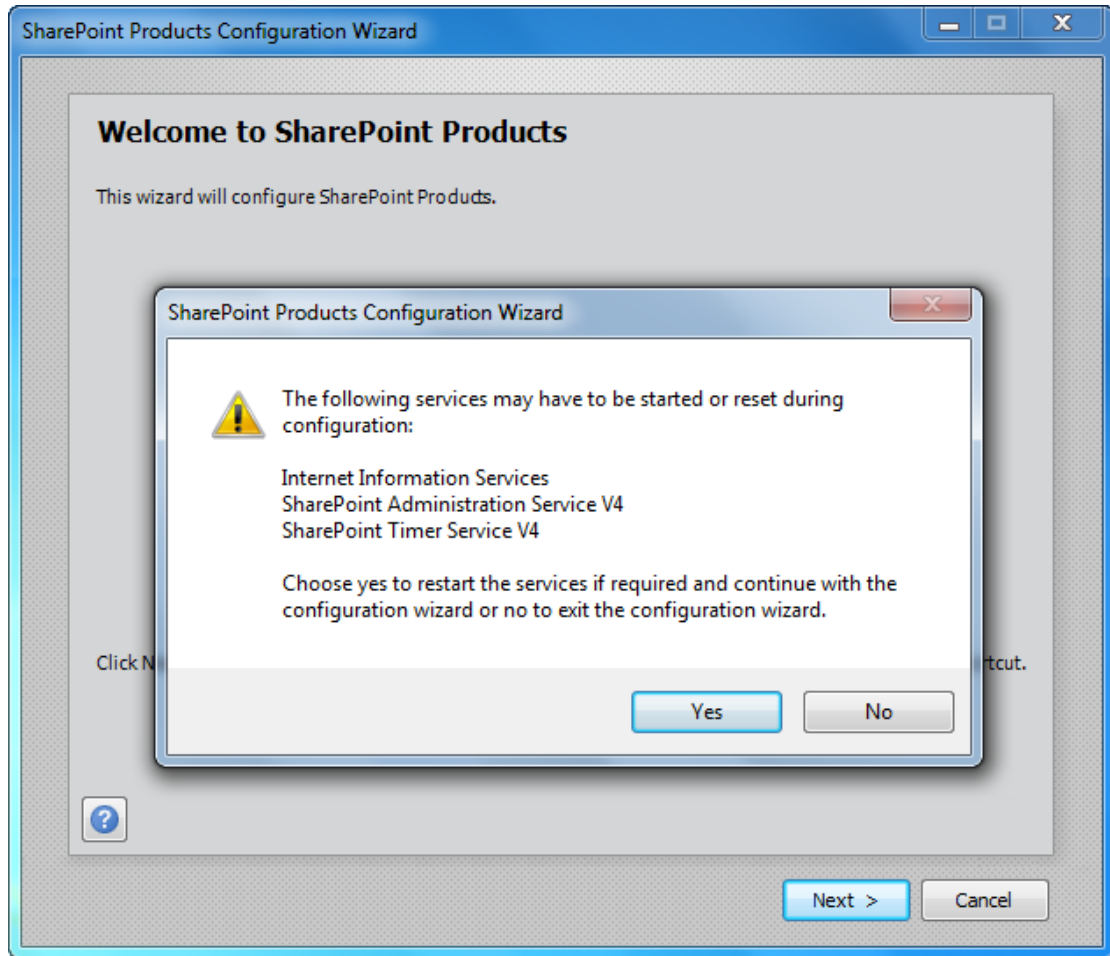
弹出的配置界面如下所示：



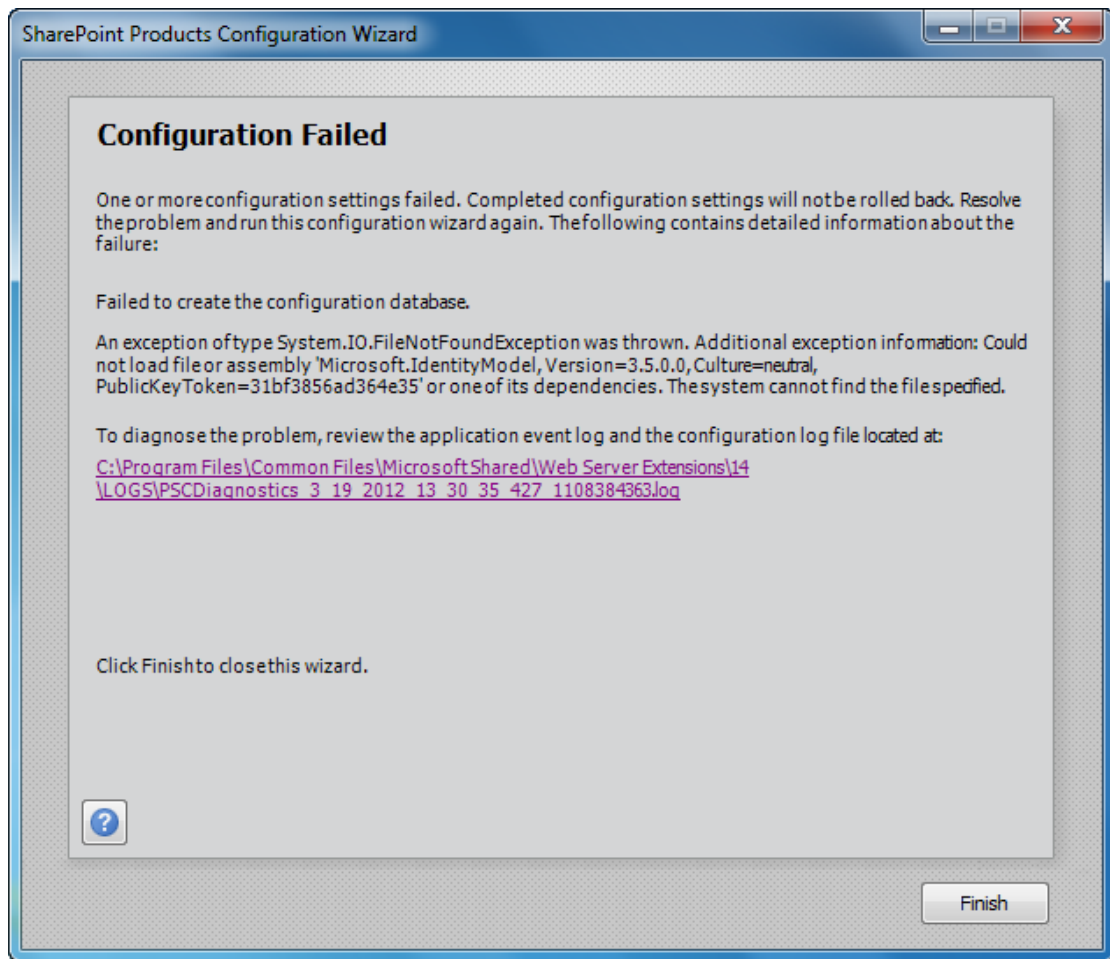
点击“Next，下一步”，出现 SharePoint 产品配置向导的警告信息，界面如下所示：



点击“OK”，直接跳过此警告，并点击下一步，弹出服务重置的警告信息，窗口如下所示：



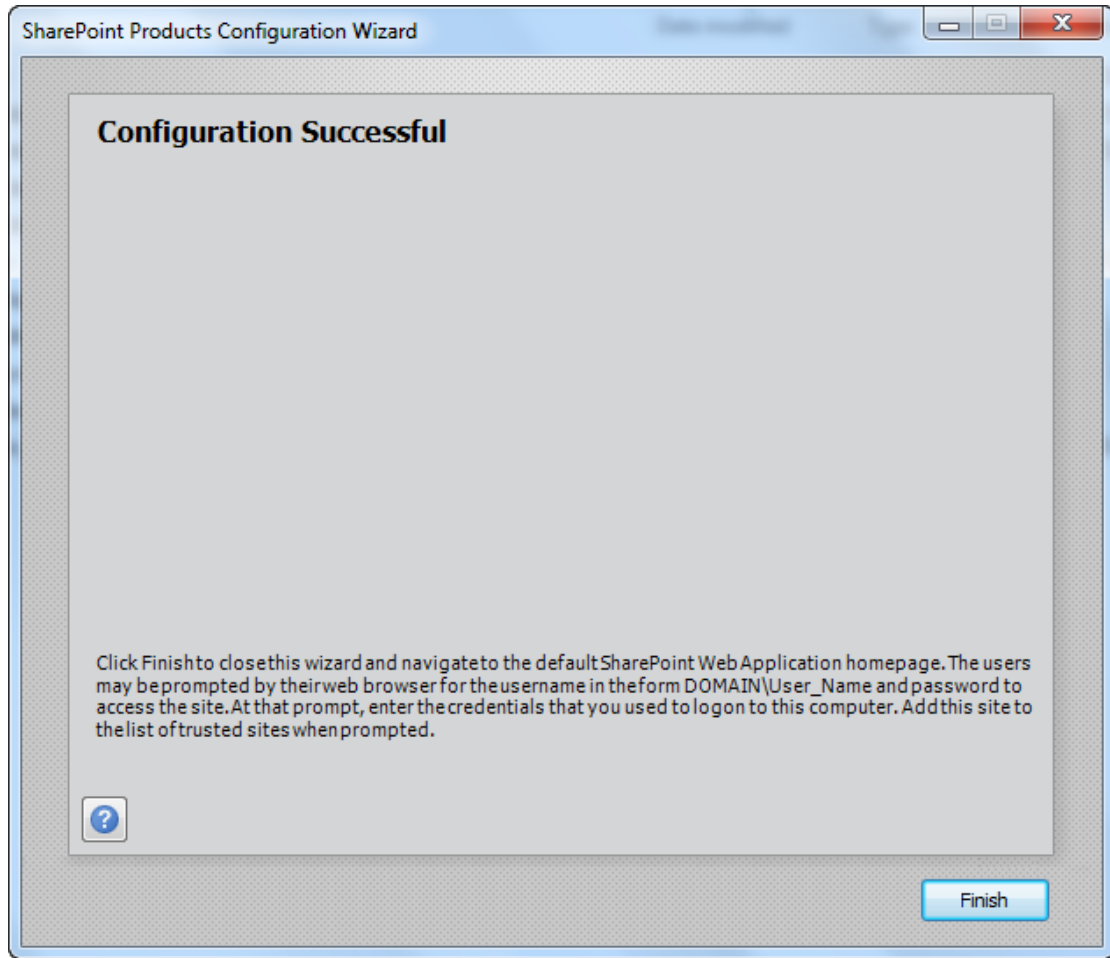
点击“OK”，直接跳过此警告，并点击下一步。出现错误信息如下图所示：



这些安装失败的原因大致是因为没有安装必备组件或者必备组件安装不正确所导致, 细心一点就可以了。

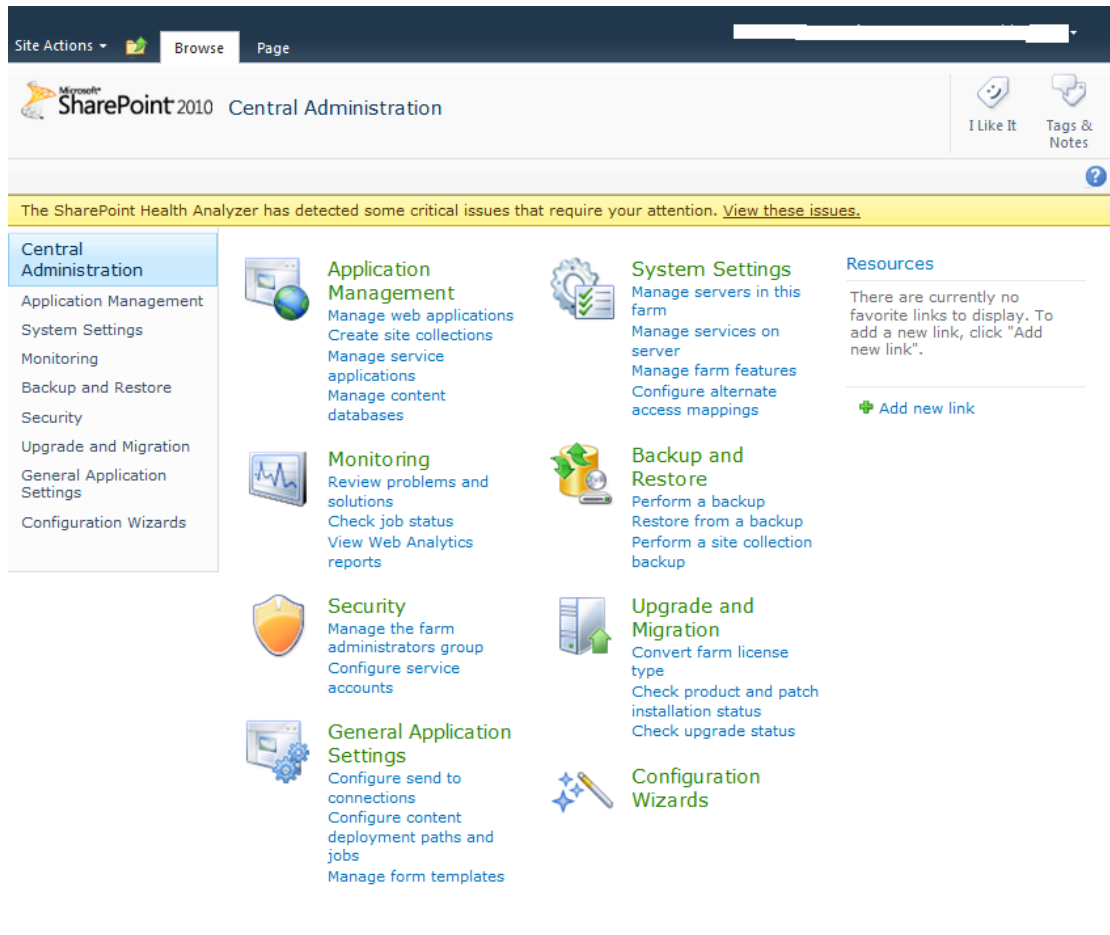
经过大约半个小时的等待以后, 配置终于成功了, 弹出的提示界面如下图所示:



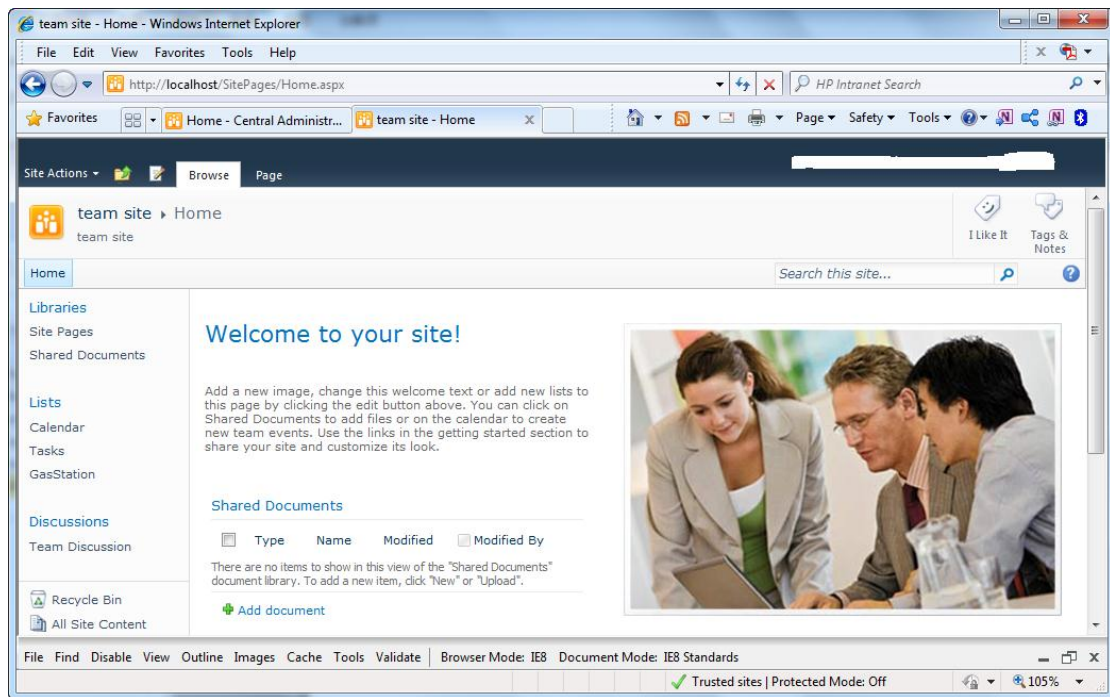


### 3.5 管理中心和站点集

点击“Finish, 完成”以后，打开管理中心的窗口如下图所示：



在管理中心创建新的 Team Site 站点集，页面如下所示：



至此，我们在 Windows 7 环境下独立安装 SharePoint Server 2010 的任务就完成了。

## 参考

[http://msdn.microsoft.com/en-us/library/ee554869\(office.14\).aspx](http://msdn.microsoft.com/en-us/library/ee554869(office.14).aspx)