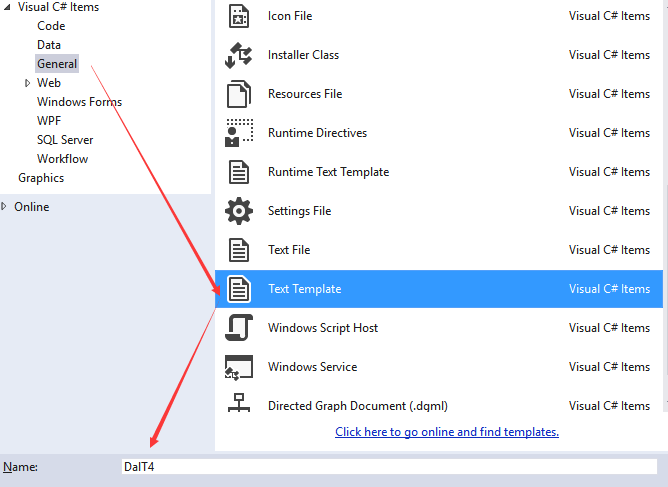
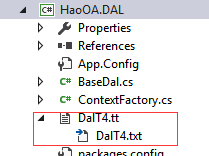
# T4创建

## Dal层

创建T4模板：

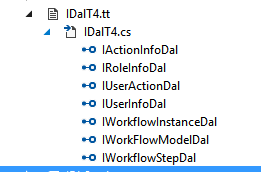


文件如下：



|  |
| --- |
| <#@ template language="C#" debug="false" hostspecific="true" #>  <#@ include file="EF.Utility.CS.ttinclude"#>  <#@ output extension=".cs" #>  <#  CodeGenerationTools code = new CodeGenerationTools(this);  MetadataLoader loader=new MetadataLoader(this);  CodeRegion region = new CodeRegion(this,1);  MetadataTools ef = new MetadataTools(this);  string inputFile = @"..\\HaoOA.Model\UserInfo.edmx";  EdmItemCollection ItemCollection = loader.CreateEdmItemCollection(inputFile);  string namespaceName = code.VsNamespaceSuggestion();  EntityFrameworkTemplateFileManager fileManager = EntityFrameworkTemplateFileManager.Create(this);  #>  using HaoOA.Model;  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace HaoOA.IDAL  {  <#  foreach(EntityType item in ItemCollection.GetItems<EntityType>().OrderBy(e=>e.Name))  {  #>  public partial interface I<#=item.Name#>Dal:IBaseDal<<#=item.Name#>>  {    }  <#}#>  } |

保存文本模板即可自动生成：



## IDbSession

|  |
| --- |
| <#@ template debug="false" hostspecific="false" language="C#" #>  <#@ include file="EF.Utility.CS.ttinclude"#>  <#@ output extension=".cs" #>  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace HM13OA.IDAL  {  public partial interface IDbSession  {  <#  MetadataLoader loader=new MetadataLoader(this);  EdmItemCollection collection= loader.CreateEdmItemCollection("HM13OA.Model/HM13OA.edmx");  foreach(EntityType item in collection.GetItems<EntityType>())  {  #>  I<#=item.Name#>Dal Get<#=item.Name#>Dal();  <#}#>  }  } |

## Dal

|  |
| --- |
| <#@ template language="C#" debug="false" hostspecific="true" #>  <#@ include file="EF.Utility.CS.ttinclude"#>  <#@ output extension=".cs" #>  <#  CodeGenerationTools code = new CodeGenerationTools(this);  MetadataLoader loader=new MetadataLoader(this);  CodeRegion region = new CodeRegion(this,1);  MetadataTools ef = new MetadataTools(this);  string inputFile = @"..\\HaoOA.Model\UserInfo.edmx";  EdmItemCollection ItemCollection = loader.CreateEdmItemCollection(inputFile);  string namespaceName = code.VsNamespaceSuggestion();  EntityFrameworkTemplateFileManager fileManager = EntityFrameworkTemplateFileManager.Create(this);  #>  using HaoOA.IDAL;  using HaoOA.Model;  using System;  using System.Collections.Generic;  using System.Data.Entity;  using System.Linq;  using System.Linq.Expressions;  using System.Text;  using System.Threading.Tasks;  namespace HaoOA.DAL  {  <#  foreach(EntityType item in ItemCollection.GetItems<EntityType>().OrderBy(e=>e.Name))  {  #>  public partial class <#=item.Name#>Dal:BaseDal<<#=item.Name#>>  {    }  <#}#>  } |

## DbSession

|  |
| --- |
| <#@ template language="C#" debug="false" hostspecific="true" #>  <#@ include file="EF.Utility.CS.ttinclude"#>  <#@ output extension=".cs" #>  <#  CodeGenerationTools code = new CodeGenerationTools(this);  MetadataLoader loader=new MetadataLoader(this);  CodeRegion region = new CodeRegion(this,1);  MetadataTools ef = new MetadataTools(this);  string inputFile = @"..\\HaoOA.Model\UserInfo.edmx";  EdmItemCollection ItemCollection = loader.CreateEdmItemCollection(inputFile);  string namespaceName = code.VsNamespaceSuggestion();  EntityFrameworkTemplateFileManager fileManager = EntityFrameworkTemplateFileManager.Create(this);  #>  using HaoOA.DAL;  using HaoOA.IDAL;  using System;  using System.Collections.Generic;  using System.Data.Entity;  using System.Linq;  using System.Reflection;  using System.Text;  using System.Threading.Tasks;  namespace HaoOA.DalFactory  {  public partial class DbSession:IDbSession  {  <#  foreach(EntityType item in ItemCollection.GetItems<EntityType>().OrderBy(e=>e.Name))  {  #>    public I<#=item.Name#>Dal Get<#=item.Name#>Dal()  {  string AC = System.Configuration.ConfigurationManager.AppSettings["<#=item.Name#>Dal"];  string[] aandc = AC.Split(',');  Assembly assembly = Assembly.Load(aandc[0]);  return assembly.CreateInstance(aandc[1]) as I<#=item.Name#>Dal;  }    <#}#>  }  } |

## IServices

|  |
| --- |
| <#@ template language="C#" debug="false" hostspecific="true" #>  <#@ include file="EF.Utility.CS.ttinclude"#>  <#@ output extension=".cs" #>  <#  CodeGenerationTools code = new CodeGenerationTools(this);  MetadataLoader loader=new MetadataLoader(this);  CodeRegion region = new CodeRegion(this,1);  MetadataTools ef = new MetadataTools(this);  string inputFile = @"..\\HaoOA.Model\UserInfo.edmx";  EdmItemCollection ItemCollection = loader.CreateEdmItemCollection(inputFile);  string namespaceName = code.VsNamespaceSuggestion();  EntityFrameworkTemplateFileManager fileManager = EntityFrameworkTemplateFileManager.Create(this);  #>  using HaoOA.Model;  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace HaoOA.IService  {  <#  foreach(EntityType item in ItemCollection.GetItems<EntityType>().OrderBy(e=>e.Name))  {  #>  public partial interface I<#=item.Name#>Service:IBaseService<<#=item.Name#>>  {    }  <#}#>  } |

## Services

|  |
| --- |
| <#@ template language="C#" debug="false" hostspecific="true" #>  <#@ include file="EF.Utility.CS.ttinclude"#>  <#@ output extension=".cs" #>  <#  CodeGenerationTools code = new CodeGenerationTools(this);  MetadataLoader loader=new MetadataLoader(this);  CodeRegion region = new CodeRegion(this,1);  MetadataTools ef = new MetadataTools(this);  string inputFile = @"..\\HaoOA.Model\UserInfo.edmx";  EdmItemCollection ItemCollection = loader.CreateEdmItemCollection(inputFile);  string namespaceName = code.VsNamespaceSuggestion();  EntityFrameworkTemplateFileManager fileManager = EntityFrameworkTemplateFileManager.Create(this);  #>  using HaoOA.IDAL;  using HaoOA.Model;  using System;  using System.Collections.Generic;  using System.Linq;  using System.Linq.Expressions;  using System.Text;  using System.Threading.Tasks;  using HaoOA.DalFactory;  using HaoOA.IService;  namespace HaoOA.Service  {  <#  foreach(EntityType item in ItemCollection.GetItems<EntityType>().OrderBy(e=>e.Name))  {  #>  public partial class <#=item.Name#>Service:BaseService<<#=item.Name#>>,I<#=item.Name#>Service  {  protected override IBaseDal<<#=item.Name#>> GetDal()  {  return dbsession.Get<#=item.Name#>Dal();  }  }  <#}#>  } |