

IS2009 制作 Oracle 静默安装包

第一部分：前言

目标需求

实现 ESL——嵌入式软件许可

在安装过程中，需要安装我们的软件产品，并且安装 Oracle 数据库。安装数据库之后需要导入基础数据，创建指定的库，创建用户和表空间。在安装 Oracle 数据时，必须实现静默安装，不能出现 GUI 界面，也不能暴露出安装参数。所有安装过程一次完成，无需用户干预。

环境说明

操作系统——Windows XP SP3

安装工具——InstallShield 2009 — Premier Edition

数据库 ——Oracle 10G-R2

目录

第一部分：前言	1
目标需求	1
环境说明	1
目录	2
第二部分：配置 Oracle.....	3
录制 response 文件.....	3
创建用户、表空间、导入基础数据.....	4
制作带基础数据的模板.....	6
安装脚本描述.....	7
集合 Oracle 安装脚本.....	10
卸载脚本描述.....	11
集合卸载脚本.....	12
第三部分：使用 IS2009	13
向导设置	13
安装设计	15
第四部分：IS2009 中的脚本编程	19
传递源路径参数.....	19
执行批处理文件.....	23
第五部分：其他	25
修改界面风格.....	25
编译打包	25
第六部分：遇到的问题.....	28
第七部分：总结	28
部分源代码	29

第二部分：配置 Oracle

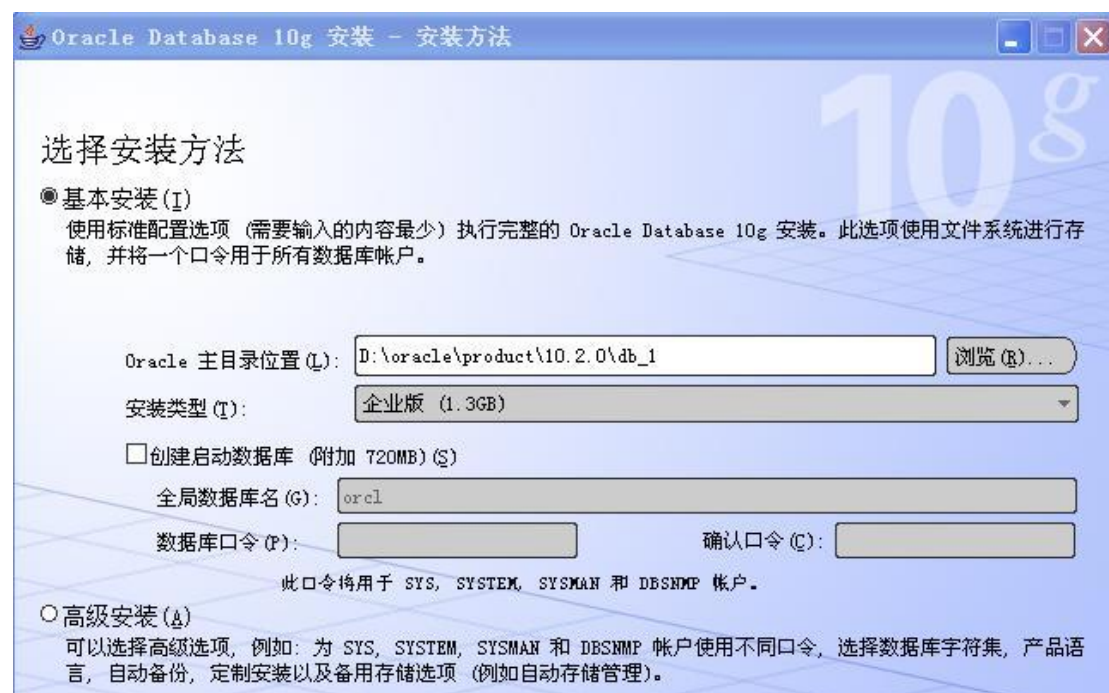
此过程需要对 Oracle 进行相关的配置，包括录制 Response 文件、制作带数据的模板、调整安装脚本、解锁用户脚本，设置环境变量等操作。

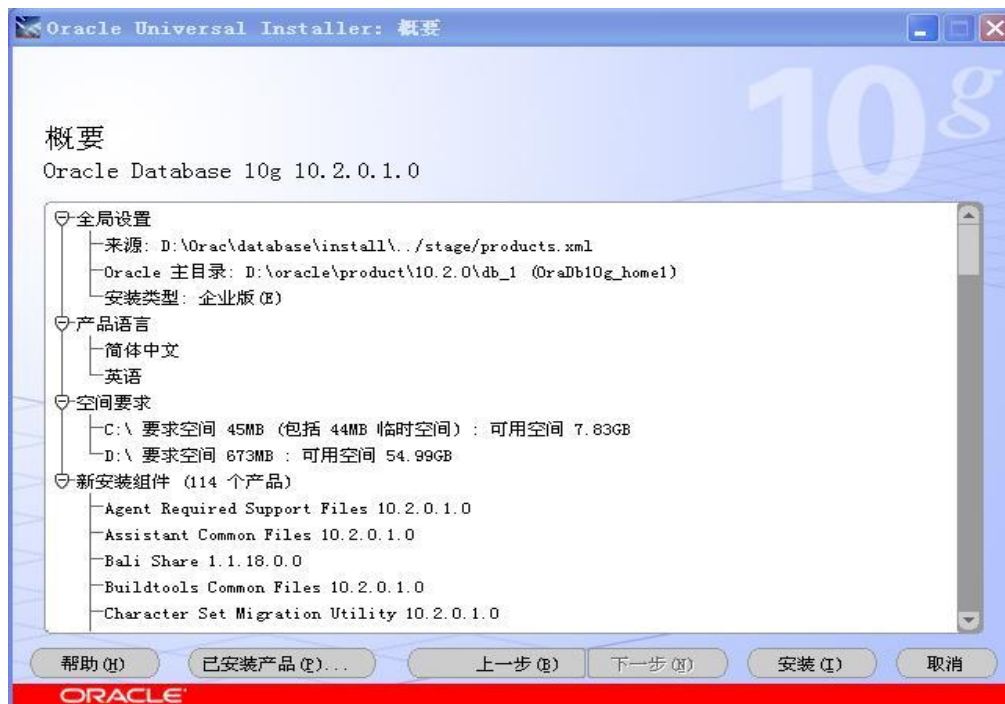
录制 response 文件

1. 运行 CMD，在命令行窗口进入 Oracle 所在路径，这里路径为 D:\Orac\database，Oracle10gserver.rsp 是要保存的文件名。

```
D:\Orac\database>setup -record -destinationfile d:\orac\database\oracle10gserver.rsp
```

2. 运行此命令后，Oracle 安装界面被打开。
3. 去掉创建启动数据库
4. 在汇总信息出现时，退出安装界面，不进行安装。



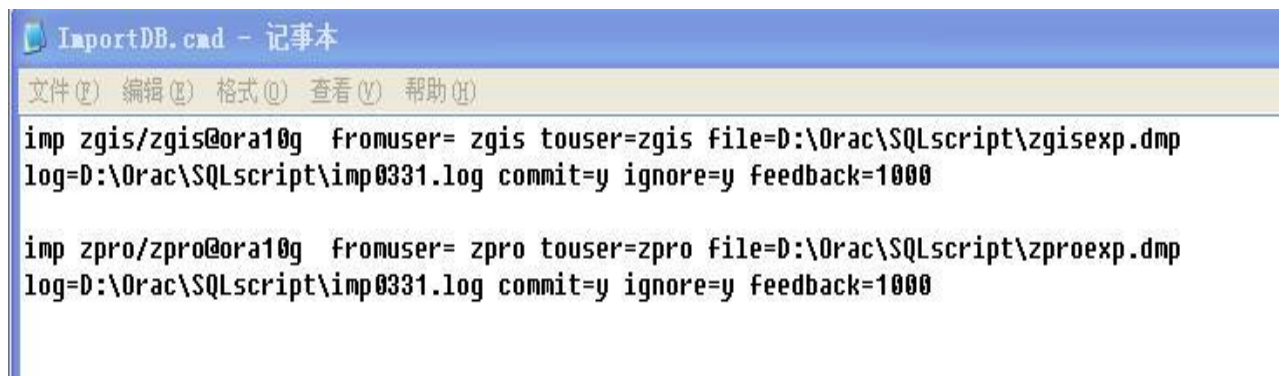


5. 此时发现在 D:\Orac\database 目录中出现了 Oracle10gserver.rsp 文件，说明录制成功。该文件存放的路径，我们后续将会在 Install.bat 中用到。

创建用户、表空间、导入基础数据

1. 在已有的同版本数据库中创建用户、表空间、导入基础数据
2. 使用 SQL 系统账号登陆，并执行 CreateDB.sql 脚本。注意：脚本中的路径应该以实际的数据路径为准。
3. 导入文件 zgisexp.dmp 和 zproexp.dmp。把此脚本存为 CMD 文件，直接执行即可。

注意：脚本中的路径应该为实际 dmp 文件存放的路径。另外此脚本为本单位的数据信息，可不作为参考！



创建用户、表空间脚本:

```
CreatedB.sql - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

--创建ZGIS表空间
CREATE SMALLFILE TABLESPACE "ZGIS"
DATAFILE 'D:\PRODUCT\10.2.0\ORADATA\SURVEY\ZGIS.DBF'
SIZE 200M
LOGGING EXTENT MANAGEMENT
LOCAL SEGMENT SPACE MANAGEMENT AUTO ;
--创建ZPRO表空间
CREATE SMALLFILE TABLESPACE "ZPRO"
DATAFILE 'D:\PRODUCT\10.2.0\ORADATA\SURVEY\ZPRO.DBF'
SIZE 500M
LOGGING EXTENT MANAGEMENT
LOCAL SEGMENT SPACE MANAGEMENT AUTO;
--删除 和 用户
DROP USER "ZGIS" CASCADE;
DROP USER "ZPRO" CASCADE;
--创建ZGIS用户
CREATE USER "ZGIS"
PROFILE "DEFAULT"
IDENTIFIED BY "ZGIS"
DEFAULT TABLESPACE "ZGIS" TEMPORARY TABLESPACE "TEMP"
ACCOUNT UNLOCK;
--授予ZGIS用户相应权限
GRANT CREATE ANY SEQUENCE TO "ZGIS";
GRANT CREATE ANY TABLE TO "ZGIS";
GRANT ENQUEUE ANY QUEUE TO "ZGIS";
GRANT UNLIMITED TABLESPACE TO "ZGIS";

GRANT "CONNECT" TO "ZGIS";
GRANT "DBA" TO "ZGIS";
GRANT "RECOVERY_CATALOG_OWNER" TO "ZGIS";
GRANT "RESOURCE" TO "ZGIS";
--创建ZPRO用户
CREATE USER "ZPRO"
PROFILE "DEFAULT"
IDENTIFIED BY "ZPRO"
DEFAULT TABLESPACE "ZPRO" TEMPORARY TABLESPACE "TEMP"
ACCOUNT UNLOCK;
--授予ZPRO用户相应权限
GRANT CREATE ANY SEQUENCE TO "ZPRO";
GRANT CREATE ANY TABLE TO "ZPRO";
GRANT ENQUEUE ANY QUEUE TO "ZPRO";
GRANT UNLIMITED TABLESPACE TO "ZPRO";

GRANT "CONNECT" TO "ZPRO";
GRANT "DBA" TO "ZPRO";
GRANT "RECOVERY_CATALOG_OWNER" TO "ZPRO";
GRANT "RESOURCE" TO "ZPRO";
```

制作带基础数据的模板

有了以上的数据，我们就开始制作数据的模板，以便建库的时候使用。

1. 创建数据库模板

在命令行下运行：

```
createCloneTemplate.cmd <ORACLE_HOME> <SID>  
    <templateName> <sysPWD>  
CreateCloneTemplate.cmd d:\oracle\product\10.2.0\db_1 ora10g  
    temp_10g oracle
```



```
time /t  
SETLOCAL  
SET MY_ORACLE_HOME=%1  
SET MY_ORACLE_SID=%2  
SET TEMPLATE_NAME=%3  
SET SYS_PASSWORD=%4  
  
%MY_ORACLE_HOME%\bin\dbca -silent -createCloneTemplate -sourceDB %MY_ORACLE_SID% -sysDBAUserName  
sys -sysDBAPassword %SYS_PASSWORD% -templateName %TEMPLATE_NAME% -maintainFileLocations false  
time /t
```

2. 创建好后，我们会在<ORACLE_HOME>\assistants\dbca\templates\目录中，发现如下两个模板

文件。**temp_10g.dbc temp_10.dfb**

此处两个文件的作用是，按照数据库模板的形式创建数据库，无需在导入表，创建用户和导入基础数据。这一步，先在已有的 Oracle 环境下完成。我们拷贝这两个文件即可。

3. 修改 temp_10g.dbc

```
<MiscParams>  
    <databaseType>MULTIPURPOSE</databaseType>  
    <percentageMemTOSGA>70</ percentageMemTOSGA >  
    <customSGA>>false</ customSGA >  
</ MiscParams >
```

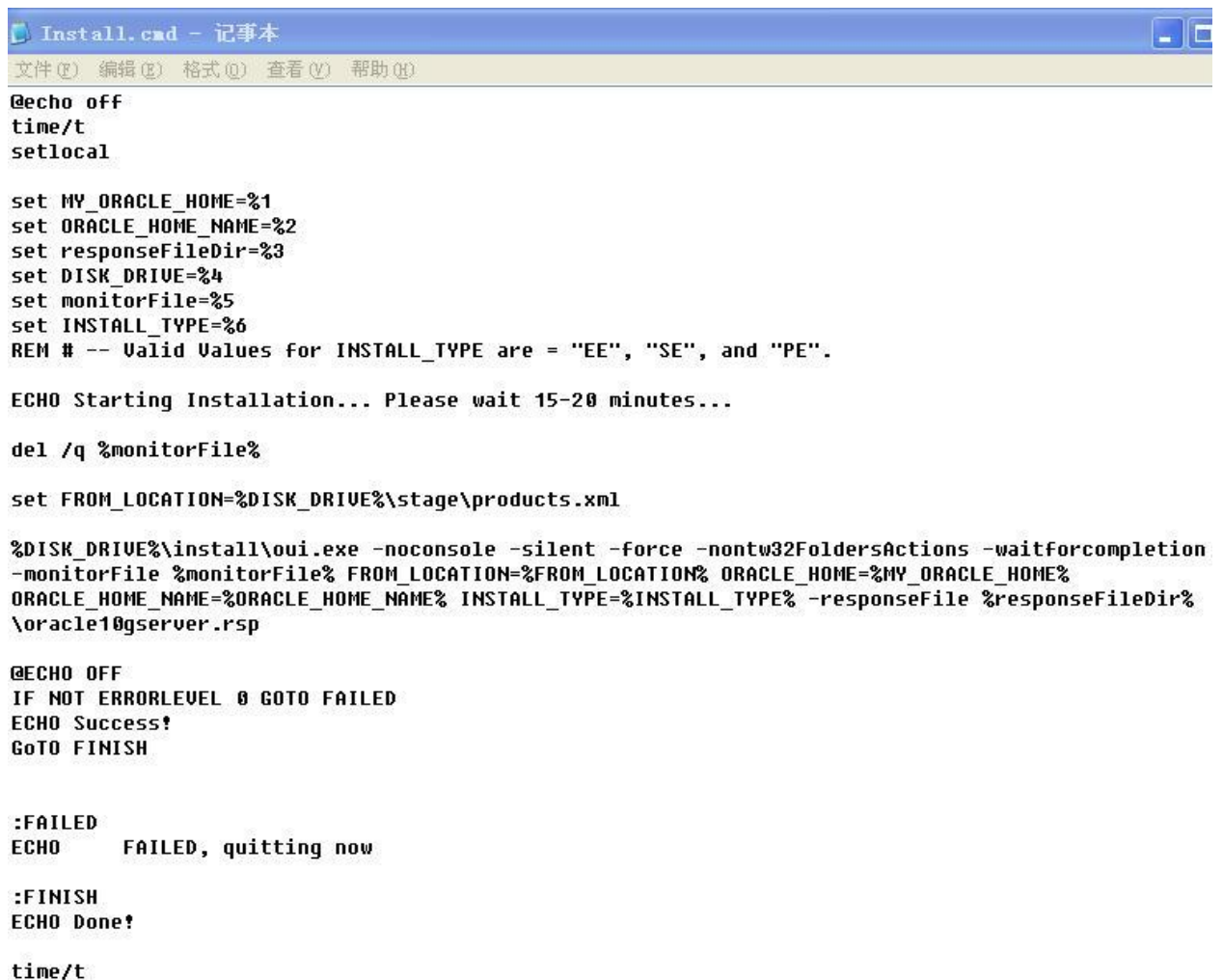

安装脚本描述

在进行 Oracle 安装过程中，需要安装程序调用以下的脚本，以实现 Oracle 的安装过程。

1. Install.cmd 安装 Oracle 软件脚本

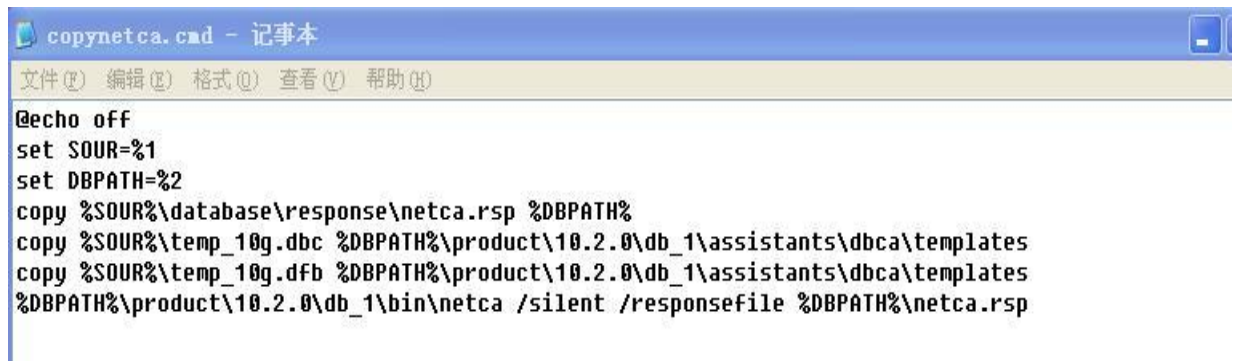
```
Install.cmd <ORACLE_HOME> <HOME_NAME>  
          <responsefileDIR> <stageDIR> <monitorFile> <installType>
```

执行此脚本时，需要带以上的参数，具体参数后边会有说明。



```
Install.cmd - 记事本  
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)  
  
@echo off  
time/t  
setlocal  
  
set MY_ORACLE_HOME=%1  
set ORACLE_HOME_NAME=%2  
set responseFileDir=%3  
set DISK_DRIVE=%4  
set monitorFile=%5  
set INSTALL_TYPE=%6  
REM # -- Valid Values for INSTALL_TYPE are = "EE", "SE", and "PE".  
  
ECHO Starting Installation... Please wait 15-20 minutes...  
  
del /q %monitorFile%  
  
set FROM_LOCATION=%DISK_DRIVE%\stage\products.xml  
  
%DISK_DRIVE%\install\oui.exe -noconsole -silent -force -nontw32FoldersActions -waitforcompletion  
-monitorFile %monitorFile% FROM_LOCATION=%FROM_LOCATION% ORACLE_HOME=%MY_ORACLE_HOME%  
ORACLE_HOME_NAME=%ORACLE_HOME_NAME% INSTALL_TYPE=%INSTALL_TYPE% -responseFile %responseFileDir%  
\oracle10gserver.rsp  
  
@ECHO OFF  
IF NOT ERRORLEVEL 0 GOTO FAILED  
ECHO Success!  
GOTO FINISH  
  
:FAILED  
ECHO      FAILED, quitting now  
  
:FINISH  
ECHO Done!  
  
time/t
```

2. copynetca.cmd 拷贝监听文件和模板文件，并创建监听服务脚本。



```
@echo off
set SOUR=%1
set DBPATH=%2
copy %SOUR%\database\response\netca.rsp %DBPATH%
copy %SOUR%\temp_10g.dbc %DBPATH%\product\10.2.0\db_1\assistants\dbca\templates
copy %SOUR%\temp_10g.dfb %DBPATH%\product\10.2.0\db_1\assistants\dbca\templates
%DBPATH%\product\10.2.0\db_1\bin\netca /silent /responsefile %DBPATH%\netca.rsp
```

Ø 脚本中存在两个接收的变量，这个是从 IS 中传出来的路径参数，必须存在。另外，模板文件必须拷贝到指定的路径下，<Oracle_Home>\assistants\dbca\templates。

3. createDB.cmd 创建带模板数据的数据库脚本。

此脚本用于根据数据模板创建数据库，命令带如下的参数：

```
createDB.cmd <ORACLE_HOME> <templateName> <SID>
               <GlobalDBName> <sysPwd> <systemPWD> <recqverArea>
               <configEM> <dbsnmpPWD> <sysmanPWD>
```



```
createDB.cmd - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

@echo off
time /t
SETLOCAL

SET MY_ORACLE_HOME=%1
REM -- provide full name of the template including the .extension like General_Purpose.dbt
SET TEMPLATE_NAME=%2
SET MY_ORACLE_SID=%3
SET MY_GLOBAL_DB_NAME=%4
SET sysPassword=%5
SET systemPassword=%6
SET recoveryAreaDestination=%7
SET CONFIGURE_EM=%8
SET dbsnmpPassword=%9

REM -- The shift command allows a argument (%10) to become %9; so windows can read its value to
desired variable.
REM -- you may need to add more shifts, you decide to pass more parameters.

SHIFT

SET sysmanPassword=%9

SET PATH=%MY_ORACLE_HOME%\bin;%PATH%

if "%configure_em%"=="NO" goto no_em_config

dbca -silent -createDatabase -templateName %TEMPLATE_NAME% -gdbname %MY_GLOBAL_DB_NAME% -sid %
MY_ORACLE_SID% -sysPassword %sysPassword% -systemPassword %systemPassword% -
recoveryAreaDestination %recoveryAreaDestination% -emConfiguration LOCAL -dbsnmpPassword %
dbsnmpPassword% -sysmanPassword %sysmanPassword%
goto end

:no_em_config
dbca -silent -createDatabase -templateName %TEMPLATE_NAME% -gdbname %MY_GLOBAL_DB_NAME% -sid %
MY_ORACLE_SID% -sysPassword %sysPassword% -systemPassword %systemPassword% -
recoveryAreaDestination %recoveryAreaDestination%

:end
```

4. unlock.bat 用户解锁脚本

安装模板生成的数据库，其用户状态为锁定，需要执行解锁脚本。

```
unlock.bat - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

@echo off
set sqlp=%1
%sqlp%\product\10.2.0\db_1\BIN\sqlplus sys/oracle as sysdba @change.sql
```

其中 change.sql 为：

```
change.sql - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
ALTER USER zpro IDENTIFIED BY zpro ACCOUNT UNLOCK;
ALTER USER zgis IDENTIFIED BY zgis ACCOUNT UNLOCK;
exit;
```

集合 Oracle 安装脚本

以上的 4 个脚本是安装过程必备的 4 个脚本，我们现在整理为一个批处理文件 install.bat，以便我们在制作安装包的时候调用该批处理。根据上述脚本的参数类型，我们把参数补完整。

```
install.bat - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
@echo off
echo 准备更新环境变量。。。

set DATABASEDIR=%1

echo 显示路径参数。。。

echo 源路径为%SOURPATH%

echo 目标路径为%DATABASEDIR%

Echo 准备安装数据库,请稍等。。。
Call %SOURPATH%\win32\install.cmd %DATABASEDIR%\product\10.2.0\db_1 Oradb10g_Home1 %SOURPATH%\database %SOURPATH%\database %DATABASEDIR%\install.log EE
Echo 安装完毕!

Echo 准备创建监听。。。
call %SOURPATH%\win32\copynetca.cmd %SOURPATH% %DATABASEDIR%
Echo 创建监听完毕!

Echo 准备创建数据库,请稍等。。。
Call %SOURPATH%\win32\createDB.cmd %DATABASEDIR%\product\10.2.0\db_1 temp_10g.dbc ora10g ora10g oracle oracle %DATABASEDIR%\recover_area YES oracle oracle
Echo 创建数据库完毕!

Echo 解锁用户并修改密码。。。
Call %SOURPATH%\win32\unlock.bat %DATABASEDIR%
Echo 成功解锁用户!

Echo *****数据库全部安装完毕! ! ! *****
```

- Ø 脚本中存在一个接收的环境变量%DATABASEDIR%，这是由 IS 运行此程序带出来的参数，此参数为安装的目标路径。
- Ø %SOURPATH%这个参数虽然没有在 BAT 文件中有任何的接收说明，但是在执行安装过程中，IS 会创建该参数在 BAT 文件头端，该参数为目标源路径地址。如何在 IS 中自动创建该变量，在第三部份中说明。
- Ø 以上的脚本分别调用了上边的 4 个脚本，他们后边均带有参数。

Ok,现在已经完成了 Oracle 方面的设置,我们需要保留的文件为 1+4 个批处理文件、1 个.SQL 文件和两个模板文件。这些文件将会在 IS 或安装中被调用。

卸载脚本描述

1. deinstall.cmd 卸载 oracle 软件, 该脚本的参数如下:

```
deinstall.cmd <ORACLE_HOME> <HOME_NAME> <responseFileDir>  
               <monitorFile>
```



```
deInstall.cmd - 记事本  
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)  
  
@echo off  
time/t  
setlocal  
  
set MY_ORACLE_HOME=%1  
set ORACLE_HOME_NAME=%2  
set responseFileDir=%3  
  
set PATH=%ORACLE_HOME%\bin;%PATH%  
net stop Oracle%ORACLE_HOME_NAME%TNSListener  
winutils\removeService Oracle%ORACLE_HOME_NAME%TNSListener  
  
%MY_ORACLE_HOME%\oui\bin\setup.exe -noconsole -silent -deinstall -waitforcompletion  
ORACLE_HOME=%MY_ORACLE_HOME% ORACLE_HOME_NAME=%ORACLE_HOME_NAME% REMOVE_HOMES="{ "%  
MY_ORACLE_HOME%}" -responseFile %responseFileDir%\oracle10gserver.rsp  
  
rd /q /s %MY_ORACLE_HOME%  
  
time/t
```

2. completeClean.cmd 清除相关文件和设置, 该脚本的相关参数如下:

```
completeClean.cmd <recoverArea> <OracleBase>  
                  <OracleInventory>
```



```
completeClean.cmd - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

time/t
setlocal

set RECOVERY_AREA=%1
set MY_ORACLE_BASE=%2
set ORACLE_INVENTORY=%3
set SOURPATH=%4

rd /s /q %RECOVERY_AREA%
rd /s /q %ORACLE_INVENTORY%
rd /s /q %MY_ORACLE_BASE%

winutils\removeORADBA
REM -- Uncomment this section only if you want to remove Oracle registry for all Oracle software
from this machine.
winutils\removeOracleHomeRegistry

del OracleRegistry.backup
time/t
```

集合卸载脚本

- Ø 以上两个脚本为卸载数据库所用，我们把他集合为一个批处理，以方便 IS 调用。
- Ø 该脚本中依然存在%DATABASEDIR%和%SOURPATH%两个参数。作用同 install.bat



```
deinstall.bat - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

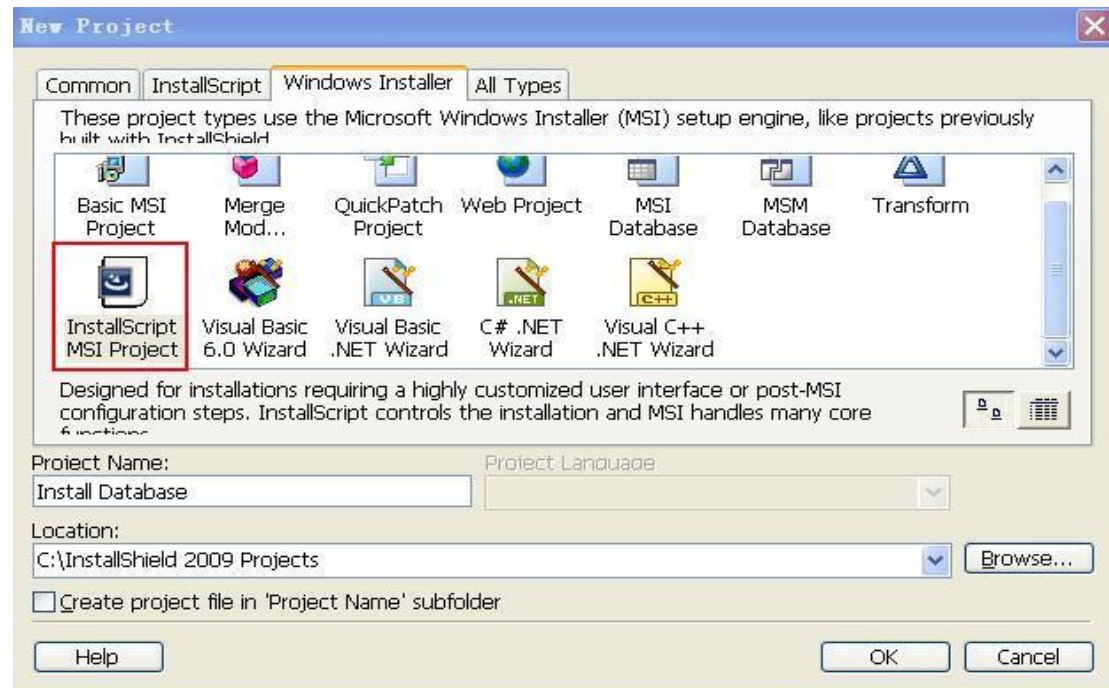
set DATABASEDIR=%1
echo 准备卸载数据库软件,请稍等...
call %SOURPATH%\win32\deinstall.cmd %DATABASEDIR%\product\10.2.0\db_1 Oradb10g_Home1 %SOURPATH%
\database %DATABASEDIR%\install.log
echo 卸载完毕...

echo 准备清除相关文件及设置...
call %SOURPATH%\win32\completeClean.cmd %DATABASEDIR%\recover_area %DATABASEDIR% "c:\program
files\oracle"
echo 清除完毕!
```

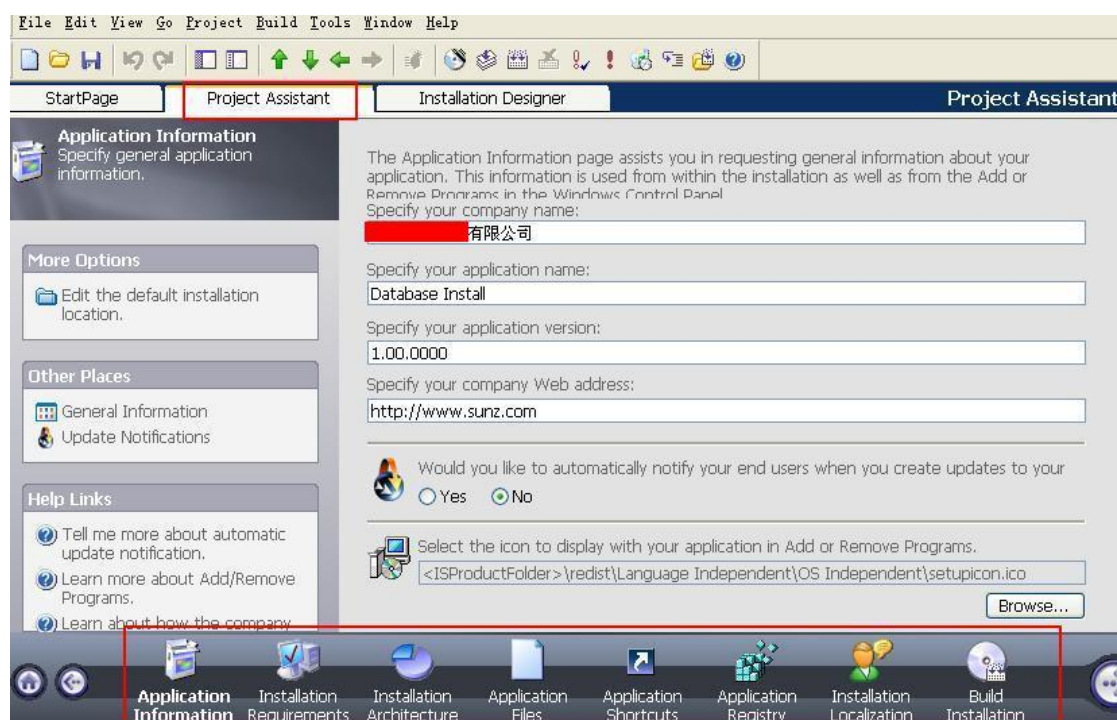

第三部分：使用 IS2009

向导设置

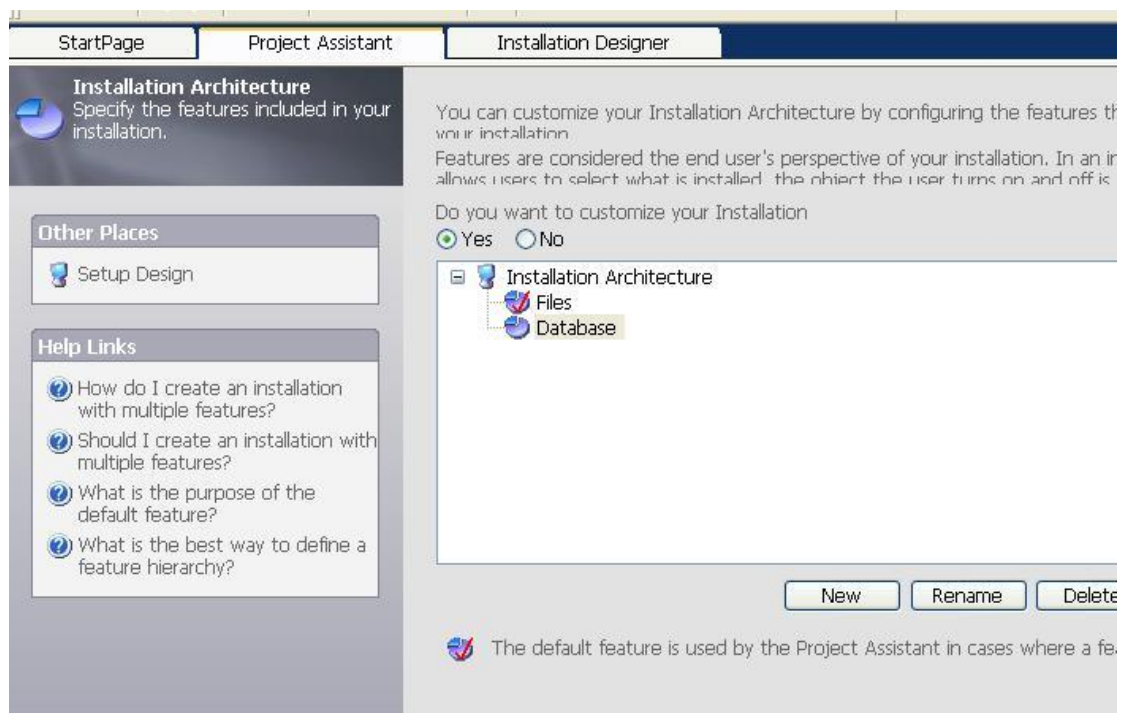
1. 打开 InstallShield2009，新建一个 Installscript MSI Project，建议选择此工程，因为该工程支持脚本编写。选择类型为 Windows Installer > InstallScript MSI Project



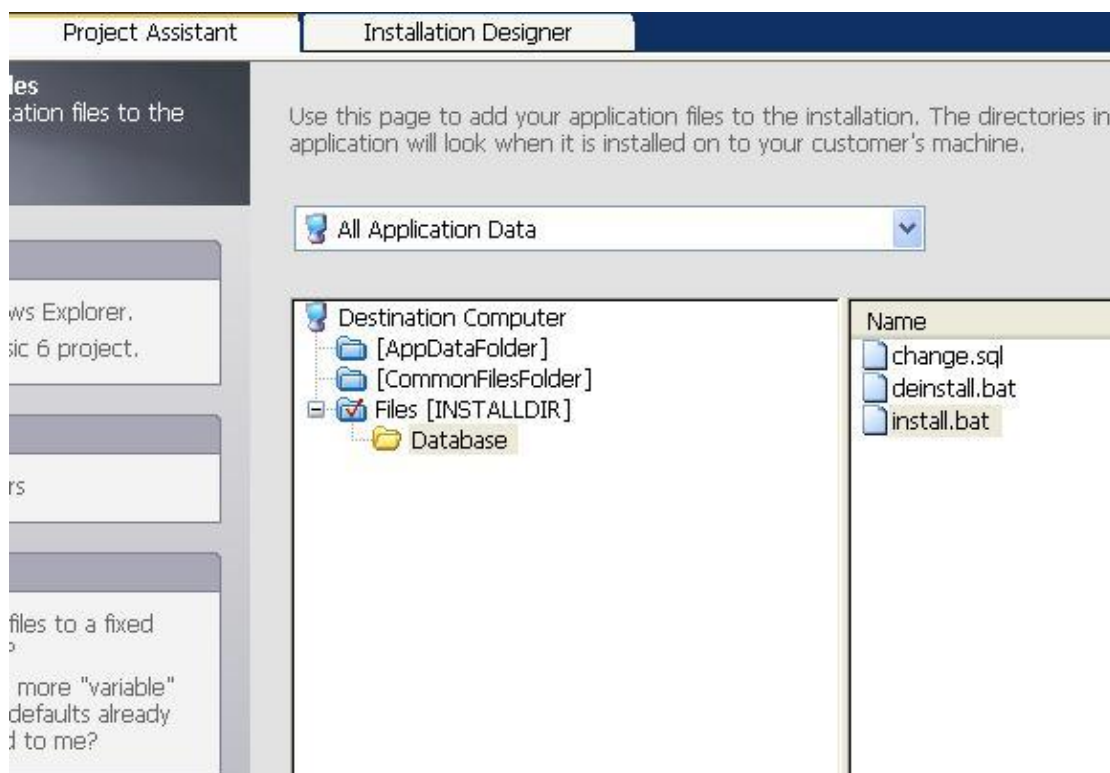
2. 进入 IS 向导设置界面，下面我们分别进行设置。下图为 Application Information 页面



3. 点击导航，进入 **Installation Architecture** 页面，选择 **YES** 设置软件组件结构。**Files** 为安装文件，**Database** 为数据库文件。



4. 点击导航，进入 **Application Files** 页面，设置安装目标文件夹，并且在 **database** 文件夹下添加如下文件。

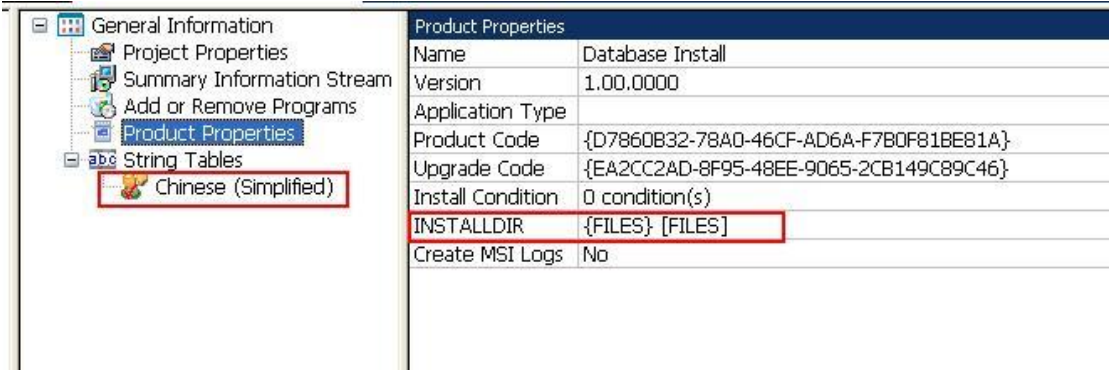


安装设计

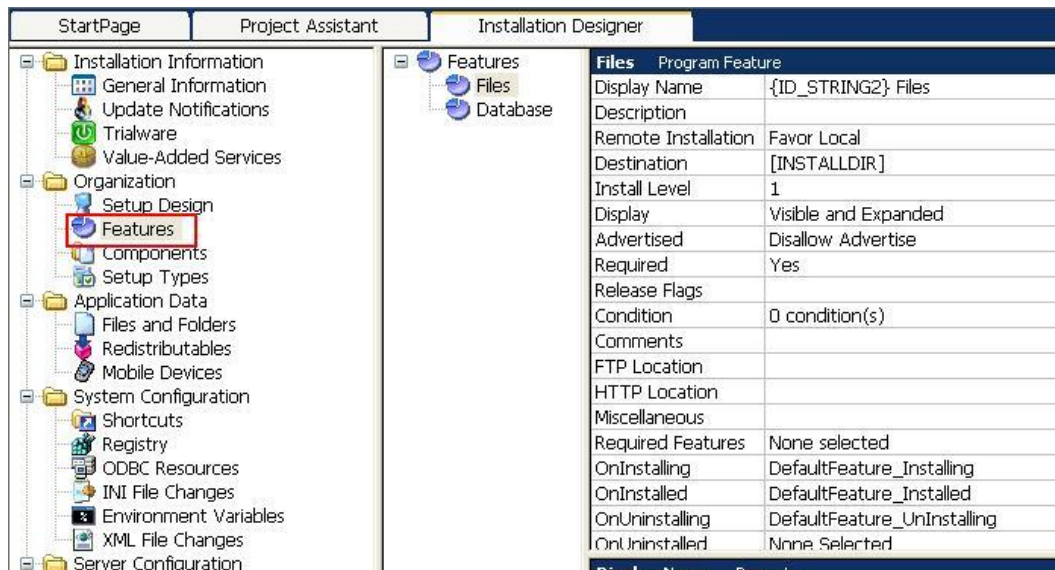
1. 在向导设置中进行简单操作后，进入安装设计标签页面。



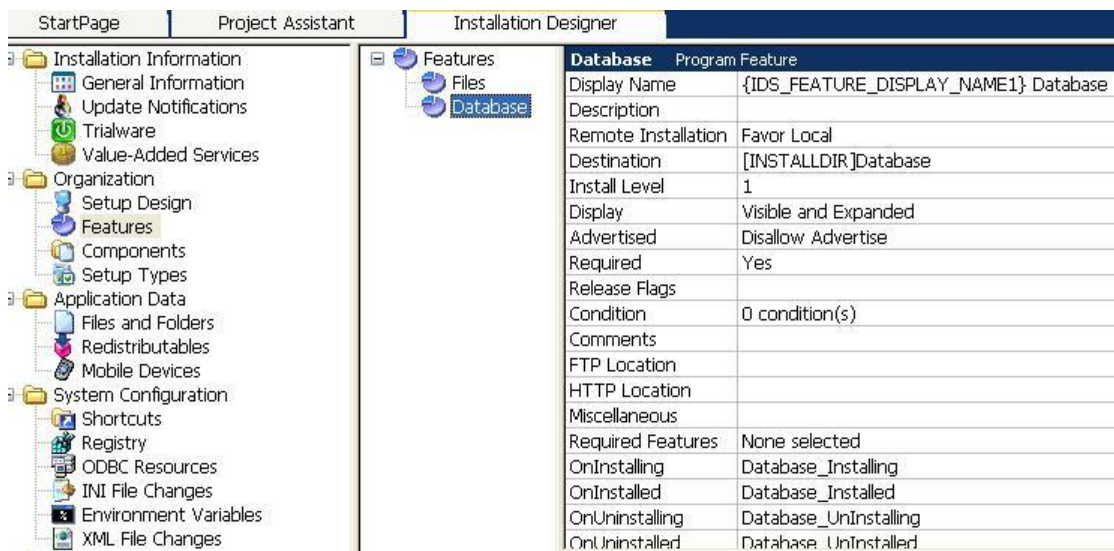
2. 在基本设置页面查看安装目录和语言设置



3. 在 Organization ->Features 中设置安装路径，以及其他的设置。下图为 Files 的设置

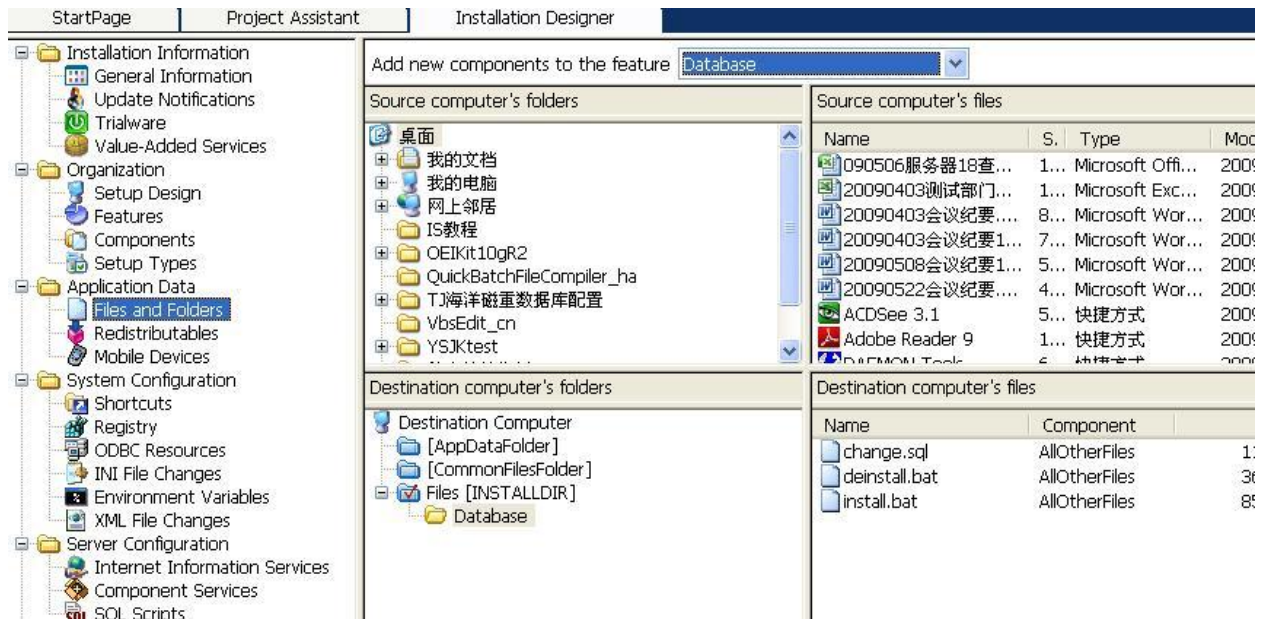


下图为 Database 的设置, 注意目标路径为[INSTALLDIR]Database, 该文件夹是在 Files 文件夹下的目录。

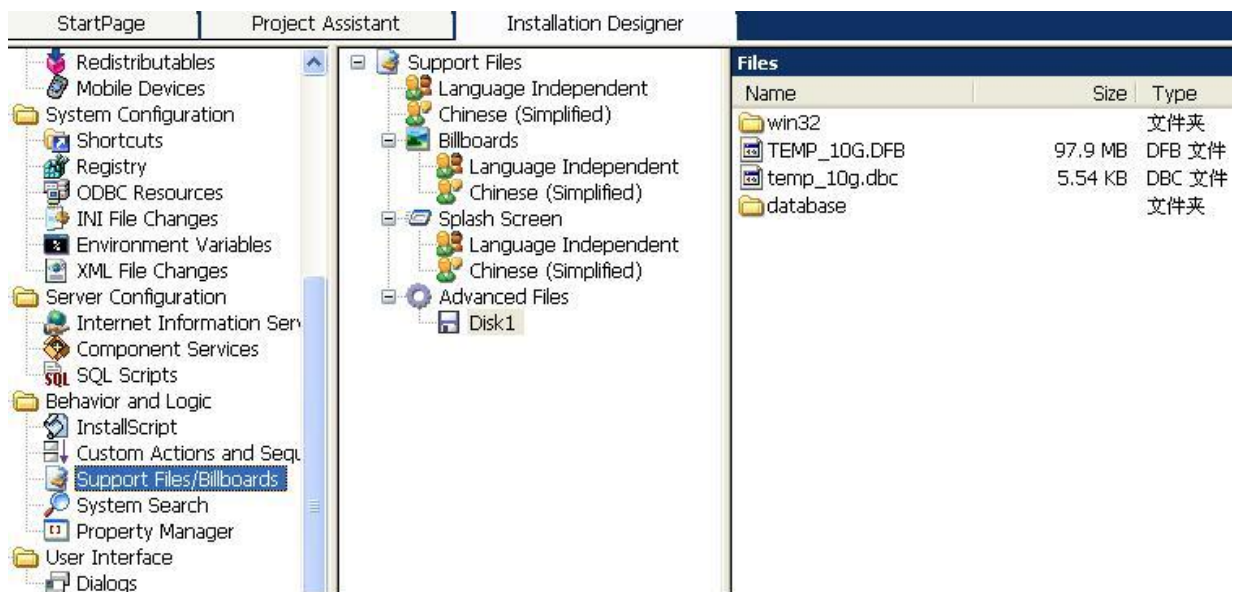


请设置好 OnInstalling、OnInstalled、OnUninstalling、OnUninstalled 选项。该选项允许用户指定和调用相关的 Install Scripe 函数。后边我们在 IS 中的脚本编程中将讲到如何在这些函数中写代码。

4. 用户也可以在 Application Data ->Files and Folders 中查看或添加修改文件。



5. 在 Behavior and Logic—>Support Files/Billboards 中的 Disk1 下添加需要安装包支持的文件和文件夹。Win32 文件夹是存放基础脚本的文件夹；database 文件夹是 Oracle 的安装文件夹；Temp_10G.dbc 和 Temp_10G.dfb 为建库模板文件。



以上步骤完成了 IS 中的基本设置和安装设置，只要是目录结构的选择，Files 文件夹是安装程序文件的文件夹，它的参数是[INSTALLDIR]。它的目录下是 Database 文件夹。参数是[INSTALLDIR]Database。 另外注意，3 个脚本文件必须放在指定的文件夹位置，这关系到脚本语言中的相互调用。

支持文件将存放在光盘的跟目录下，这给我们直接调用这些文件带来了方便。

第四部分：IS2009 中的脚本编程

传递源路径参数

用户在使用 IS 安装过程中，我们不能够将所有信息都复制到本地硬盘进行安装，通常情况下会读取源盘上的一些信息，譬如光盘下的某个路径中的内容。但是如何获取安装的路径信息呢？又如何让批处理文件获知这些内容呢？

这里我们用到 IS 函数库中的 3 个函数：

BatchAdd（添加一个环境变量到一个批处理文件）

BatchFileLoad（把一个批处理文件装入内存来用高级批处理函数编辑它）

BatchFileSave（保存一个由 **BatchFileLoad** 装入的批处理文件）

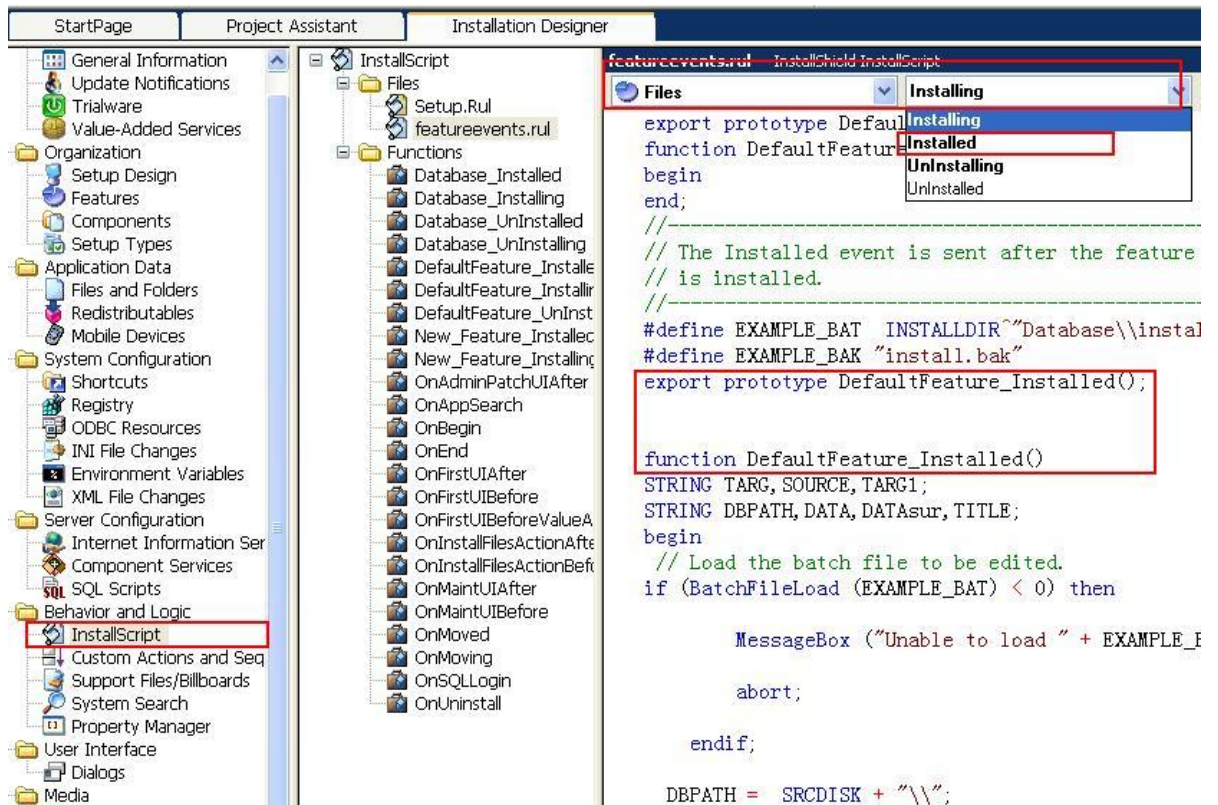
BatchAdd 函数插入一个 SET 命令或其它 DOS 命令到一个已经由 **BatchFileLoad** 装入内存的批处理文件中。参数 **nOptions** 使你可以将新命令添加到文件的第一个或最后一个语句，用新命令取代一个现存语句，或指定将新命令添加到一个现存语句的前面或后面。调用 **BatchAdd** 前，你必须调用 **BatchFileLoad** 来把要修改的文件装入内存。在你修改该文件后，调用 **BatchFileSave** 来把它保存到磁盘。

下面我们在 IS2009 中去使用这些函数。

1. 在 Installation Designer 中选择 IS 左边栏的 Behavior and Logic 中的 InstallScript。
2. 在脚本框的上方，下拉选择框，选择 Files
3. 之后，选择右边的下拉框，选择 Installed

之所以选择 **Installed**，是因为安装过程中会把 **Install.bat** 批处理脚本复制到本地硬盘，只有在成功复制之后，我们才能调用该批处理脚本，使用这些函数。

如图所示：



Installed 函数中的具体脚本如下：

```
#define EXAMPLE_BAT  INSTALLDIR^"Database\\install.bat"
```

```
#define EXAMPLE_BAK "install.bak"
```

```
export prototype DefaultFeature_Installed();
```

```
function DefaultFeature_Installed()
```

```
STRING TARG,SOURCE,TARG1;
```

```
STRING DBPATH,DATA,DATAsur,TITLE;
```

```
begin
```

```
    // Load the batch file to be edited.
```

```
    if (BatchFileLoad (EXAMPLE_BAT) < 0) then
```

```
        MessageBox ("Unable to load " + EXAMPLE_BAT + ". ", SEVERE);
```

```
        abort;
```



```

endif;

DBPATH = SRCDISK + "\\";

// Add the line SET PATH = SRCDISK+"\".

if (BatchAdd ("SOURPATH", DBPATH, "", BEFORE) < 0) then

    MessageBox ("Second call to BatchAdd failed", WARNING);

    abort;

endif;

// Save the updated file; back up the original file.

if (BatchFileSave(EXAMPLE_BAK) < 0) then

    MessageBox ("Unable to save " + EXAMPLE_BAK + ".", SEVERE);

//else

    // MessageBox ("Batch file saved. Backup created.", INFORMATION);

endif;

```

该脚本中，我们定义 **Install.bat** 的安装路径，并且在开始处用 **BatchFileLoad** 函数调用它，注意，该调用并不是执行它，而是在内存中临时调用。然后设置一个变量 **DBPATH = SRCDISK + "\\"**，**SRCDISK** 是 IS 判断 **Setup.inx** 所在的位置，取出安装盘符，当然我们后边要加上反斜杠。譬如光盘在 **F:** 我们这样做的目的就是在批处理的第一行插入 **Set SOURPATH = F:**。**BatchAdd** 函数用于加载这个变量，保证在该批处理文件中最开始就设置这一变量。最后使用 **BatchFileSave** 保存批处理文件。

同样，在卸载时候我们同样需要在 **Deinstall.bat** 中这样去做，只是位置要确定在 **Uninstalling**，并且定义 **deinstall.bat** 的安装路径即可。



UnInstalling 函数中的具体脚本如下：

```
#define UNINSTALL_BAT  INSTALLDIR^"Database\deinstall.bat"

#define UNINSTALL_BAK "deinstall.bak"

export prototype DefaultFeature_UnInstalling();

function DefaultFeature_UnInstalling()

STRING TARG,SOURCE;

STRING DBPATH,DATA,DATAsur,TITLE;

begin

// Load the batch file to be edited.

if (BatchFileLoad (UNINSTALL_BAT) < 0) then

    MessageBox ("Unable to load " + EXAMPLE_BAT+".", SEVERE);

    abort;

endif;

DBPATH =  SRCDISK + "\\";

// Add the line SET PATH = SRCDISK+"\\".

if (BatchAdd ("SOURPATH", DBPATH, "", BEFORE) < 0) then

    MessageBox ("Second call to BatchAdd failed", WARNING);

    abort;

endif;

// Save the updated file; back up the original file.

if (BatchFileSave(UNINSTALL_BAK) < 0) then

    MessageBox ("Unable to save " + UNINSTALL_BAK + ".", SEVERE);
```

```
//else

//  MessageBox ("Batch file saved. Backup created.",INFORMATION);

endif;
```

执行批处理文件

我们修改完批处理文件，就要在修改保存之后执行它。这需要用到 Is 函数库中的 LaunchAppAndWait () 函数

语法：LaunchAppAndWait (szProgram, szCmdLine, IWait);

说明：LaunchAppAndWait 函数运行由 szProgram 指定的带有 szCmdLine 指定的命令行参数的应用程序。第三个参数，IWait 指示安装在继续前是否要等待直到运行的应用程序终止。

一个安装程序只能监控由 szProgram 指定的应用程序；如果该应用程序要运行其它应用程序或进程，安装程序不能监控它们。因此，安装程序将在第一个应用程序结束后继续，即使那时由第一个应用程序运行的其它应用程序仍在运行。注意如果运行的应用程序终止失败，则安装程序将无限等待运行的应用程序完成。

添加在 Installed 脚本的后面：

```
TARG = INSTALLDIR^"Database";

if (LaunchAppAndWait(INSTALLDIR^"Database\install.bat",TARG,LAAW_OPTION_WAIT |
LAAW_OPTION_HIDDEN) < 0) then

    MessageBox ("Unable to launch install.bat",SEVERE);

endif;

end;
```

LAAW_OPTION_HIDDEN 这个参数是隐藏 bat 的执行窗口，它与 LAAW_OPTION_WAIT 并用是隐藏并且等待程序执行完成后返回，注意他们的前后顺序。

添加在 UnInstalling 脚本的后面：

```

TARG = INSTALLDIR^"Database";

if
(LaunchAppAndWait(INSTALLDIR^"Database\\deinstall.bat",TARG,LAAW_OPTION_WAIT
LAAW_OPTION_HIDDEN) < 0) then

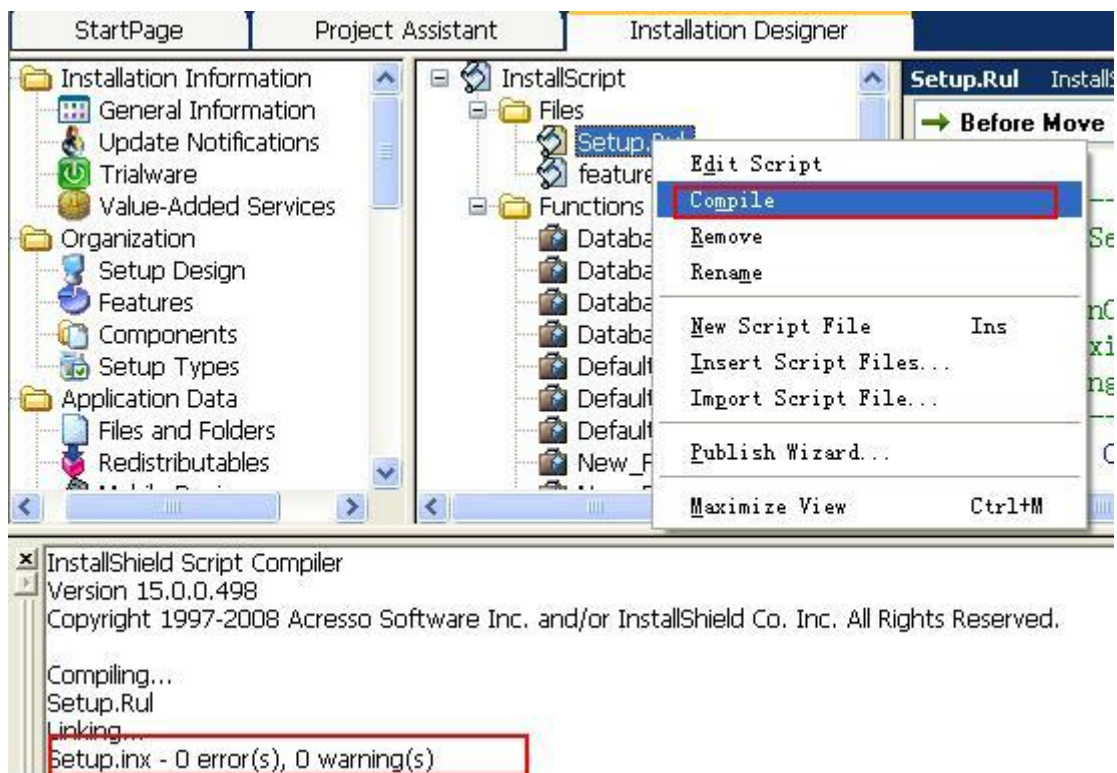
    MessageBox ("Unable to launch deinstall.bat",SEVERE);

endif;

end;

```

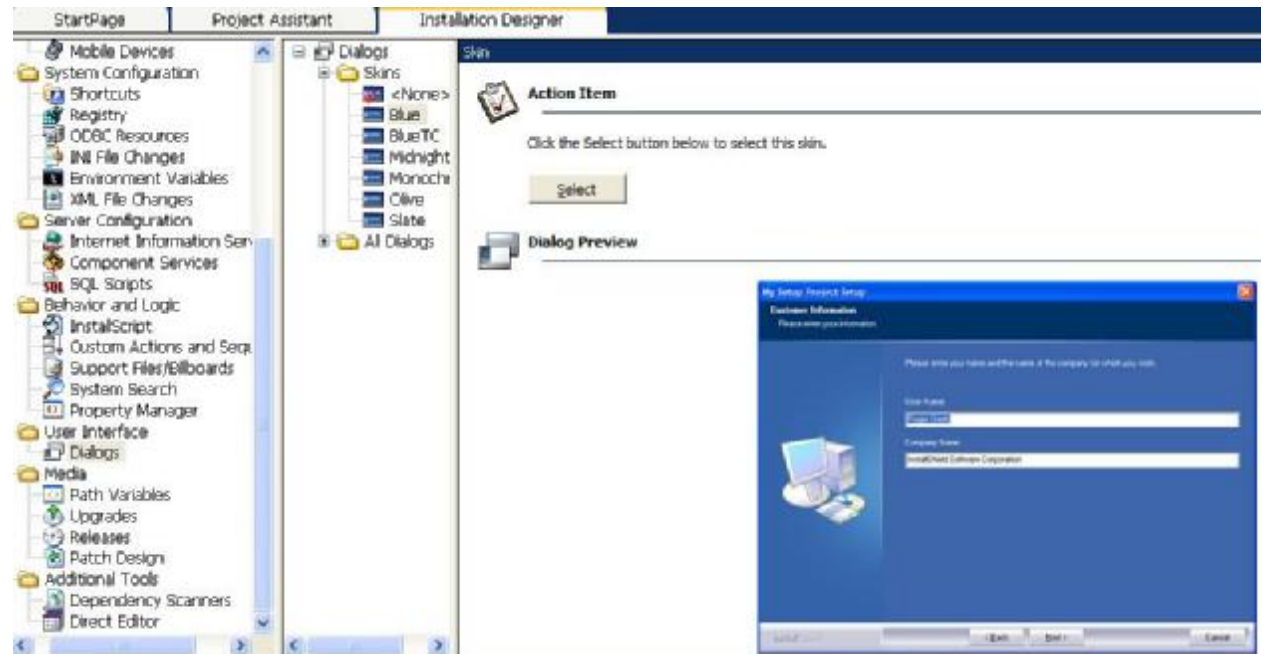
脚本已经完成，我们编译脚本，并确定没有任何的问题。



第五部分：其他

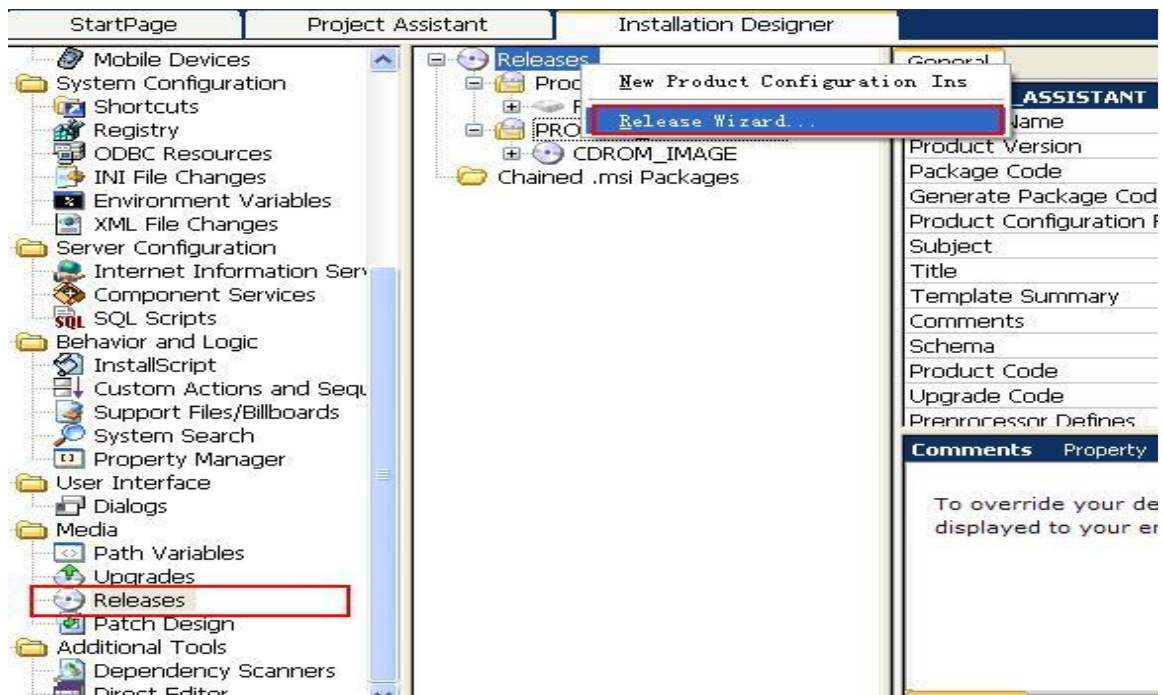
修改界面风格

1. 在 Installation Designer 中选择 IS 左边栏的 User Interface 中的 Dialogs
2. 在右边窗口中选择 Skins，选择之后点击 Select
3. 界面窗口风格改变

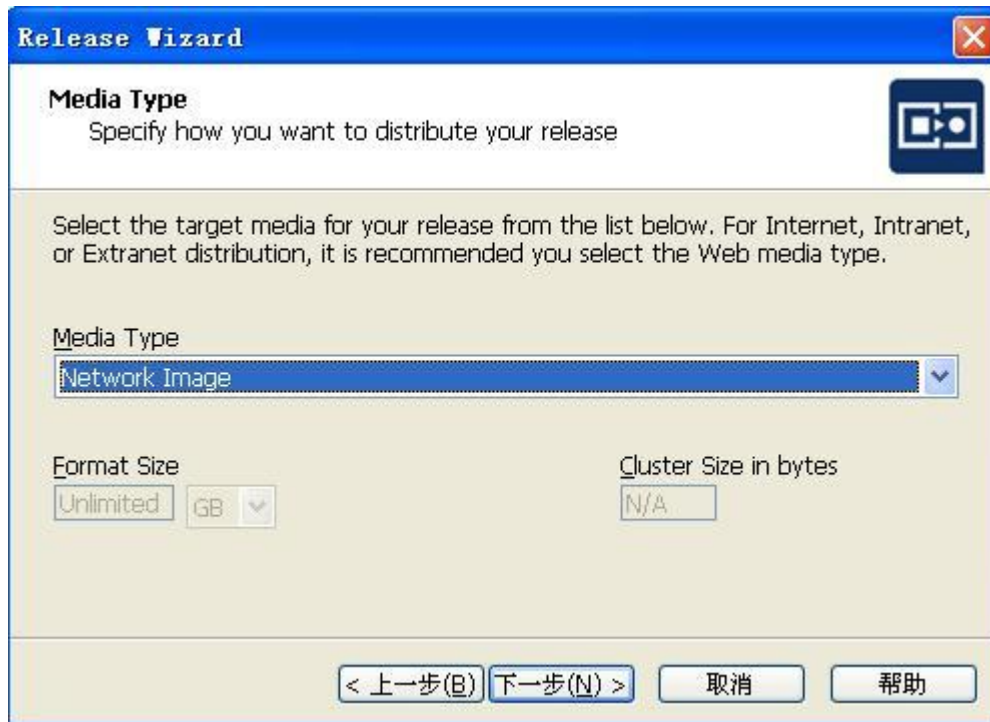


编译打包

1. 在 Installation Designer 中选择 IS 左边栏的 Media 中的 Releases
2. 在右边栏 Releases 点击右键，选择 Releases Wizard...
3. 出现向导界面
4. 在该界面中，注意几个界面，其余均为默认
5. Filter Setting 界面，不要做任何的选择
6. Media Type 界面，选择 Network Image



注意：该地点不要选择过滤!! 默认即可，若选择，导致读取光盘 1158 问题！上图为错误图



完整的安装包文件



第六部分：遇到的问题

1158 问题：过滤语言操作导致，不选择过滤项即可

中文输入乱码问题：在向导中进行中文输入即可

环境变量问题：由于 IS 提供设置环境变量的功能，先开始认为在 IS 中设置环境变量就可以通过 bat 去读取，但是这样做是错误的，变量是可以添加成功，但不允许该次的调用，所以放弃了这种方法，改用函数。

设计思路问题：先开始是创建新库，并分别导入表，创建用户，导入数据，这样做麻烦的要死，参数问题不断。后来改用数据模板建库。

目录问题：开始是这样打算的，把文件和数据库用不同的目录区分开，UI 中出现两个路径选择页面，可是 IS 中只提供一个安装 INSTALLDIR 的目录的参数，另外一个目录路径我不知道如何获取和调用。故采用在 INSTALLDIR 目录下又创建了 Database 文件夹。

第七部分：总结

本人第一次使用 IS，走了不少弯路，先开始用 IS Express 版本，发现根本无法使用脚本，改成 IS09，由于对 Oracle 的不熟悉，批处理的不熟悉，IS 编程语言的不熟悉，导致此次制作过程颇为漫长。零零散散历时一个多月的时间。

这里要感谢 IS 群的帮助，是他们给我前进的动力。问题一一排除，最终做好产品，提交 Oracle。特别感谢，海洋女神，奕婷，Blue，宇心，csp 等等。。。

以上全部内容仅仅包含 ORACLE 的打包，公司的产品应用并没有放在其中。

IS 群 19622645 空白制作 QQ: 28021418

部分源代码

```
export prototype New_Feature_Installing();
function New_Feature_Installing()
begin
end;

//-----
// The Installing event is sent before the feature DefaultFeature
// is installed.
//-----

export prototype DefaultFeature_Installing();
function DefaultFeature_Installing()

STRING diskpath,szCommand,szCmdLine,szCommandecp,szCmdLineecp;
begin
    diskpath =      SRCDISK + "\\"; //定义安装盘路径
//运行 framework20 安装程序
    if (LaunchAppAndWait(SRCDIR^"supportfiles\dotnetfx.exe","",LAAW_OPTION_WAIT/* |
LAAW_OPTION_HIDDEN*/) < 0) then
        MessageBox ("Unable to launch dotnet",SEVERE);
    endif;
    if
(LaunchAppAndWait(SRCDIR^"supportfiles\framework2.0\setup.exe","",LAAW_OPTION_WAIT/*
|
LAAW_OPTION_HIDDEN*/) < 0) then
        MessageBox ("Unable to launch dotnet2",SEVERE);
    endif;
    //定义 MSI 文件参数及 MSI 文件位置
    szCommand = WINSYSDIR^"msiexec.exe";
    LongPathToShortPath(szCommand);
    szCmdLine =SRCDIR^"supportfiles\EngineRT\setup.msi";
        LongPathToShortPath(szCmdLine);
    //运行 MSI 文件
    if (LaunchAppAndWait( szCommand,"/i"+szCmdLine,LAAW_OPTION_WAIT) < 0) then

        MessageBox ("Unable to launch ACGIS",SEVERE);
    endif;
    // szCommandecp = SRCDIR^"supportfiles\SoftwareAuthorization.exe";
    //     LongPathToShortPath(szCommandecp);
    // szCmdLineecp =SRCDIR^"supportfiles\license.ecp";
    //     LongPathToShortPath(szCmdLineecp);
```

```

        if (LaunchAppAndWait(SRCDIR^"supportfiles\\Runecp.bat",diskpath,LAAW_OPTION_WAIT/*
|
LAAW_OPTION_HIDDEN*) < 0) then

        MessageBox ("Unable to launch ecg",SEVERE);
        endif;

end;

//-----
// The Installed event is sent after the feature DefaultFeature
// is installed.
//-----
#define EXAMPLE_BAT  INSTALLDIR^"Database\\install.bat"
#define EXAMPLE_BAK "install.bak"
export prototype DefaultFeature_Installed();

function DefaultFeature_Installed()
STRING TARG,SOURCE,TARG1;
STRING DBPATH,DATA,DATAsur,TITLE;
begin
    // Load the batch file to be edited.
    if (BatchFileLoad (EXAMPLE_BAT) < 0) then

        MessageBox ("Unable to load " + EXAMPLE_BAT+".", SEVERE);

        abort;

    endif;

    DBPATH =  SRCDISK + "\\";

    // Add the line SET PATH = SRCDISK^"test".
    if (BatchAdd ("SOURPATH", DBPATH, "", BEFORE) < 0) then

        MessageBox ("Second call to BatchAdd failed", WARNING);

        abort;

    endif;

    // Save the updated file; back up the original file.

```

```

if (BatchFileSave(EXAMPLE_BAK) < 0) then

    MsgBox ("Unable to save " + EXAMPLE_BAK + ". ", SEVERE);

//else

    //  MsgBox ("Batch file saved. Backup created.", INFORMATION);

endif;


//DATAsur = "C:\\";
//DATA = SRCDISK^"test";
// TITLE = "Path buffer example";
// PathSet (DATAsur);
//SprintfBox (INFORMATION,TITLE,

//          "The starting search path is %s.", DATAsur);
// if (PathAdd(DATA, "", FULL, BEFORE) < 0) then

    //  MsgBox ("Unable to add DATA Path to path buffer.", SEVERE);

    // abort;

// endif;

// PathGet (DATAsur);
//SprintfBox (INFORMATION,TITLE,

//          "DATA Path added before first path.\n\nThe search path is %s.",

//          DATAsur);


TARG = INSTALLDIR^"Database";
//  SOURCE = SRCDISK^"test";

// LaunchAppAndWait(SRCDIR^"path.bat",SOURCE,WAIT);

```



```

    if (LaunchAppAndWait(INSTALLDIR^"Database\\install.bat",TARG,LAAW_OPTION_WAIT |
LAAW_OPTION_HIDDEN) < 0) then
        MessageBox ("Unable to launch install.bat",SEVERE);
    endif;
    //Delay(1);
    // LaunchAppAndWait(INSTALLDIR^"createdb.bat",TARG,LAAW_OPTION_WAIT);
    //Delay(2);
    // LaunchAppAndWait(SRCDISK^"test//sqlscript//setup.bat",TARG,LAAW_OPTION_WAIT);
    // Delay(2);
    // LaunchAppAndWait(SRCDISK^"test//sqlscript//importdb.bat",TARG,LAAW_OPTION_WAIT);

end;
//-----
// The UnInstalling event is sent just before the feature
// DefaultFeature is uninstalled.
//-----
#define UNINSTALL_BAT  INSTALLDIR^"Database\\deinstall.bat"
#define UNINSTALL_BAK "deinstall.bak"
export prototype DefaultFeature_UnInstalling();
function DefaultFeature_UnInstalling()
STRING TARG,SOURCE;
STRING DBPATH,DATA,DATAsur,TITLE;
begin
// Load the batch file to be edited.
if (BatchFileLoad (UNINSTALL_BAT) < 0) then

        MessageBox ("Unable to load " + EXAMPLE_BAT+".", SEVERE);

        abort;

    endif;

    DBPATH =  SRCDISK + "\\";

    // Add the line SET PATH = SRCDISK^"test".
    if (BatchAdd ("SOURPATH", DBPATH, "", BEFORE) < 0) then

        MessageBox ("Second call to BatchAdd failed", WARNING);

        abort;

    endif;

```

```

// Save the updated file; back up the original file.

if (BatchFileSave(UNINSTALL_BAK) < 0) then

    MessageBox ("Unable to save " + UNINSTALL_BAK + ". ", SEVERE);

//else

    //  MessageBox ("Batch file saved. Backup created.",INFORMATION);

endif;


//DATAsur = "C:\\";
//DATA = SRCDISK^"test";
// TITLE = "Path buffer example";
// PathSet (DATAsur);
//sprintfBox (INFORMATION,TITLE,

//
//          "The starting search path is %s.",DATAsur);
// if (PathAdd(DATA, "", FULL, BEFORE) < 0) then

//  MessageBox ("Unable to add DATA Path to path buffer.", SEVERE);

// abort;

// endif;

// PathGet (DATAsur);
//sprintfBox (INFORMATION,TITLE,

//
//          "DATA Path added before first path.\n\nThe search path is %s.",

//
//          DATAsur);


TARG = INSTALLDIR^"Database";
//  SOURCE = SRCDISK^"test";

```

```
// LaunchAppAndWait(SRCDIR^"path.bat",SOURCE,WAIT);

    if (LaunchAppAndWait(INSTALLDIR^"Database\deinstall.bat",TARG,LAAW_OPTION_WAIT |  
LAAW_OPTION_HIDDEN) < 0) then  
        MessageBox ("Unable to launch deinstall.bat",SEVERE);  
    endif;  
end;
```