Performing post-installation action items on a Siebel instance

This section includes the following topics:

About performing actions on a Siebel instance

The Interpoint for Siebel extension of Precise for SQL Server and Precise for Oracle can only receive notifications of user events if specific objects are included in the Siebel Repository File (SRF). Before you can use Interpoint for Siebel, you need to modify the SRF file. This document describes how to add the required objects to the file.

Setting up Siebel batch process sampling

For Siebel batch processes to be sampled by the Siebel AppTier Collectors, you need to create a tab-delimited text file that contains the mapping of each Siebel server to its host machine and network address. Each row in the file represents a physical machine. The first field in each row should represent the name of the Siebel server, the second field its corresponding host machine name, and the third its network address.

About sampling Siebel batch processes (SQL Server)

If you have installed Interpoint for a SQL Server database, perform the following procedures.

To create the text file

1. Log on to the database server where your Siebel AppTier Collector is installed.
2. Create the following tab-delimited text file:
   `<Precise_root>\products\sql-server\<instance_ name>\etc\siebel_machines.txt`

To obtain the list of Siebel servers

1. Log on to the SrvrMgr utility.
2. Execute the following command:
   `list servers`
   Each returned row represents a Siebel server.

To obtain the host machine name and network address

1. Copy the getAddress.vbs script from the Utilities\Siebel directory on the installation DVD to the server.
2. Run the following command:
   `cscript //nologo getAddress.vbs ODBC data source`
   `Administrator user name Administrator passwords`
   where ODBC data source is the name of the ODBC data source used by the Siebel server to connect to SQL Server. To determine its name, check the ConnectString value in the siebel.cfg configuration file.
   Administrator user name and *Administrator password* are the user name and password required to connect to the SQL Server instance.
3. Repeat these steps for each Siebel server.

Following is an example for the content of the siebel_machines.txt file:

    sieb1 APPSRV1 00096B26E66C
    sieb2 APPSRV2 00096B26E67C

Modifying the Siebel Repository File (SQL Server)

Once the Siebel AppTier Collectors are installed, modify the Siebel Repository File (SRF), so that Interpoint for Siebel will be able to receive notifications of Siebel-related user events.

Part of the .srf file, is a collection of callback scripts, which are triggered by various events in a user's session. A user can view or change these scripts using the Siebel Tools.

- For each targeted Siebel application, modify the .srf file on the Siebel server. Define the application name by hard coding it in the Application_PreNavigate function.
- The modified SRFs need to be distributed to each of the Siebel servers

If you have installed Interpoint for Siebel for a SQL Server database, modifying the SRF consists of the following tasks:

- Creating the Precise Interpoint project (SQL Server)

Creating the Precise Interpoint project (SQL Server)

The following procedure creates the VERITAS Interpoint project. All objects created in the procedures following on this procedure will be grouped under this project.

To create the Interpoint project

1. Open Siebel Tools.
2. In the Object Explorer, on the Types tab, select **Project**.
3. Create a new project by pressing Ctrl+n.
4. Name the new project, "Precise Interpoint."
5. Lock the project by clicking the **Locked** column to the right.

Creating the CX_VTSSPID table (SQL Server)

The following procedure creates a table in the DBMS.

To create the CX_VTSSPID table

1. From the File menu, select **New Object**.
2. In the New Object Wizard, on the General tab, select **Table**; then click **OK**.
3. In the General dialog box, do the following:
    a. In the Enter a name for the new Table text box, type **CX_VTSSPID**.
    b. From the Choose a Project in which you wish to create the Table list, select **Precise Interpoint**.
    c. Under Select the type of the Table, select **A stand-alone Table**.
4. Click **Next**; then click **Finish** to close the dialog box.
5. In the Tables workspace, click **Apply**.
6. In the Apply Schema dialog box, if needed, type a valid privileged user ID and privileged user password into the relevant text boxes.
7. Click **Apply**. The dialog box closes, and the table is created.
8. In the Tables panel, click **Activate** to activate the new table.

Creating the business component (SQL Server)

The following procedure creates the VTS_SPID business component.

To create the business component

1. In Siebel Tools, from the File menu, select **New Object**. The New Object Wizard opens.
2. On the General tab, select **BusComp**.
3. Click **OK**.
4. In the New Business Component dialog box, do the following:
    a. In the Enter a name for the new Business Component text box, type **VTS_SPID**.
    b. From the Select the Project this Business Component will be part of list, select **Precise Interpoint**.
    c. From the Select the Table this Business Component will operate on list, select **CX_VTSSPID**.
5. Click **Next**.
6. In the Single Value Fields dialog box, from the Select a column in the Base table list, select **ROW_ID**. A default Field name automatically appears in the Enter a name for the Field text box.
7. Click **Add** to move the selected row to the table; then click **Finish** to create the business component.

Creating the business object (SQL Server)

The following procedure creates the VTS_SPID business object.

To create the business object

1. In Siebel Tools, in the Objects Explorer, on the Types tab, select **Business Object**.
2. In the Business Objects panel, from the Edit menu, select **New Record**. A new record is added to the table in the Business Objects panel.
3. In the Name column, type: **VTS_SPID**.
4. In the Project column, assign the new record to the VERITAS Interpoint project.
5. On the Types tab in the Object Explorer, select Business Object > Business Object Component.
6. From the Edit menu, select **New Record**. A new record is added to the table in the Business Object Components panel.
7. In the BusComp column, type: **VTS_SPID**.
8. From the Tools menu, select **Compile Project**.

Adding the Application_PreNavigate() function (SQL Server)

The following procedure adds the Application_PreNavigate() function.

To add the Application_PreNavigate() function

1. In Siebel Tools, from the View menu, select **Options**.
2. In the Development Tools Options dialog box, on the Scripting tab, note the value of the Scripting Language list and click **OK**.
3. In the Object Explorer, on the Types tab, expand Application.
4. Right-click on the application to be modified and select **Lock Object**.
5. Right-click on the application to be modified and select **Edit Server Scripts**.
6. In the Server Script Editor, select the **Application_PreNavigate** function. The function's default script appears in the text panel.
   If this function already contains a custom script, contact Precise Customer Support for alternative instructions.
7. Replace the default script with the following script:

```
Function Application_PreNavigate (DestViewName As String, DestBusObjName As String) As Integer
    Dim cmd As String
    Dim oActionBO As BusObject
    Dim vtsBC As BusComp
    Dim spid As String
    Const appName = "Siebel Sales Enterprise"
    On Error Resume Next Set oActionBO = theApplication.GetBusObject("VTS_SPID") Set vtsBC = oActionBO.
GetBusComp("VTS_SPID")
With vtsBC
    ClearToQuery
    ActivateField "Row Id"
    ExecuteQuery ForwardOnly
    FirstRecord
    i = .GetFieldValue("Row Id")
End With
    Set oActionBO = Nothing
    Set vtsBC = Nothing
    cmd = "vrts_srf_agent.exe " + i + " """ + appName + """ """ + DestViewName + """ """ + DestBusObjName
+ """ """ + LoginName + """"
    Shell(cmd)
    Application_PreNavigate = ContinueOperation
End Function
```

If the Scripting Language in use is eScript, use the following script:

```
function Application_PreNavigate (DestViewName, DestBusObjName)
{
    try
    {
            var cmd;
            var oActionBO;
            var vtsBC;
            var appName = "Siebel Sales Enterprise";
            oActionBO = TheApplication().GetBusObject("VTS_SPID");
            vtsBC = oActionBO.GetBusComp("VTS_SPID");
            vtsBC.ClearToQuery();
            vtsBC.ActivateField("Row Id");
            vtsBC.ExecuteQuery(ForwardOnly);
            vtsBC.FirstRecord();
            cmd = "start cmd /c vrts_srf_agent.exe " +
            vtsBC.GetFieldValue("Row Id") +//SPID
            " \"" + appName +
            "\" \"" + DestViewName +
            "\" \"" + DestBusObjName +
            "\"\"" + TheApplication().LoginName() +
            "\"";
        vtsBC = null;
        oActionBO = null;
        Clib.system(cmd);
    }
    catch (obj)
    {
        Clib.system( "start cmd /c vrts_srf_agent.exe \"" + obj + "\"" );
        return (ContinueOperation); //Ignore all exceptions
    }
}
```

8. Modify the appName variable to match the name of the Siebel application.

Replacing the table with a custom view (SQL Server)

The following procedure replaces the table with a custom view. To replace the table with a custom view

1. Log on to your Siebel database using Query Analyzer. Make sure to log on with a privileged account.
2. Run the following script:

```
DROP TABLE CX_VTSSPID
GO
CREATE VIEW [CX_VTSSPID] as select
    '1' as [CONFLICT_ID],
    getdate() as [CREATED],
```

```
              '--' as [CREATED_BY],
              getdate() as [LAST_UPD],
              '2' as [LAST_UPD_BY],
              1 as [MODIFICATION_NUM],
              @@spid as [ROW_ID]
         GO
         GRANT SELECT ON [CX_VTSSPID] TO public
         GO
```

Completing the modification (SQL Server)

The following procedure completes the modification.

To complete the modification

1. Save all changes.
2. Compile the Siebel project. This will create a new Siebel.srf file.
3. Test the project on your development machine.
4. On each of your production Siebel Servers:
   a. Stop the Siebel Server.
   b. Deploy the new Siebel.srf file
   c. Verify that the EnableScripting option in the application CFG file is set to TRUE.
   d. Start the Siebel Server.

Debugging the script (SQL Server)

The following procedure debugs the script.

To debug the script

1. Before attempting to debug, from the View menu, select **Options**.
2. On the Debug file tab, open the file that is specified in the CFG file text box in a text editor.
3. Make sure that the RepositoryFile and ApplicationName options are set correctly.
4. Make sure the EnableScripting option is set to TRUE.

About sampling Siebel batch processes (Oracle)

For Siebel batch processes to be sampled by the Interpoint for Siebel Collectors, create a tab-delimited text file that contains the mapping of each Siebel server to its host machine, as it appears in Oracle.

To create the text file:

1. Log on to the server where your Siebel AppTier Collector instance is installed.
2. Create the following tab-delimited text file:
   - **Windows**
     `<Precise_root>\products\oracle\<SID>\etc\siebel_machines.txt`
   - **UNIX**
     `<Precise_root>/products/oracle/<SID>/etc/siebel_machines.txt`

To obtain the list of Siebel servers:

1. Log on to the SrvrMgr utility.
2. Execute the following command:
   `list servers`

The following list is an example of the results:

Table 1 List of Siebel servers

SBLSRVR_NAME	HOST_NAME
AsDev1	POOL-SIEBEL1
AsDev2	POOL-SIEBEL2

Each returned row represents a Siebel server. The row information can be split into two columns:

- The first column, `SBLSRVR_NAME` represents the name of the Siebel server, for example `AsDev1`.
- The second column `HOST_NAME` represents the corresponding physical machine name, but not necessary as it appears in Oracle.

To find the physical machine name as it appears in Oracle, connect to the Siebel database and execute the following command:

`SELECT DISTINCT MACHINE FROM V$SESSION;`

The following list is an example of the results:

Table 2 Machine name list

MACHINE
PRECISE\POOL-SIEBEL1
PRECISE\POOL-SIEBEL2

If there is no entry in `V$SESSION` with the physical machine name, use the results from the `HOST_NAME` column. After inserting the previous results into the `siebel_machines.txt` file, the contents is as follows:

```
AsDev1 PRECISE\POOL-SIEBEL1
AsDev2 PRECISE\POOL-SIEBEL2
```

Modifying the Siebel Repository File (Oracle)

Once the Siebel AppTier Collectors are installed, modify the Siebel Repository File (SRF), so that Interpoint for Siebel will be able to receive notifications of Siebel-related user events.

Part of the .srf file, is a collection of callback scripts, which are triggered by various events in a user's session. A user can view or change these scripts using the Siebel Tools.

- For each targeted Siebel application, modify the .srf file on the Siebel server. Define the application name by hard coding it in the Application_PreNavigate function.
- The modified SRFs need to be distributed to each of the Siebel servers

If you have installed Interpoint for an Oracle database, modifying the SRF consists of the following tasks:

- Creating the Precise Interpoint project (Oracle)
- Creating the CX_PSS_SBL table (Oracle)
- Creating the business component (Oracle)
- Creating the business object (Oracle)
- Adding the Application_PreNavigate() function (Oracle)
- Replacing the table with a global temporary table (Oracle)
- Completing the modification (Oracle)
- Debugging the script (Oracle)

Creating the Precise Interpoint project (Oracle)

1. Open Siebel Tools.
2. In the Siebel Tools, on the Types tab in the Object Explorer, select **Project**.
3. Create a new project by pressing Ctrl+n.
4. Name the new project, "Precise Interpoint."
5. Lock the project, by choosing the **Locked** column to the right option.

> ⓘ All objects created in subsequent steps will be grouped under this project.

Creating the CX_PSS_SBL table (Oracle)

> ⓘ This procedure applies to versions up to version 8.0 and version 8.0 and higher. For steps where there are differences between versions, the difference has been clearly specified.

1. From the File menu, choose **New Object**. The New Object Wizard opens.
2. On the General tab, select **Table**.

3. Click **OK**. The General dialog box opens.
   Figure 1 General Dialog Box
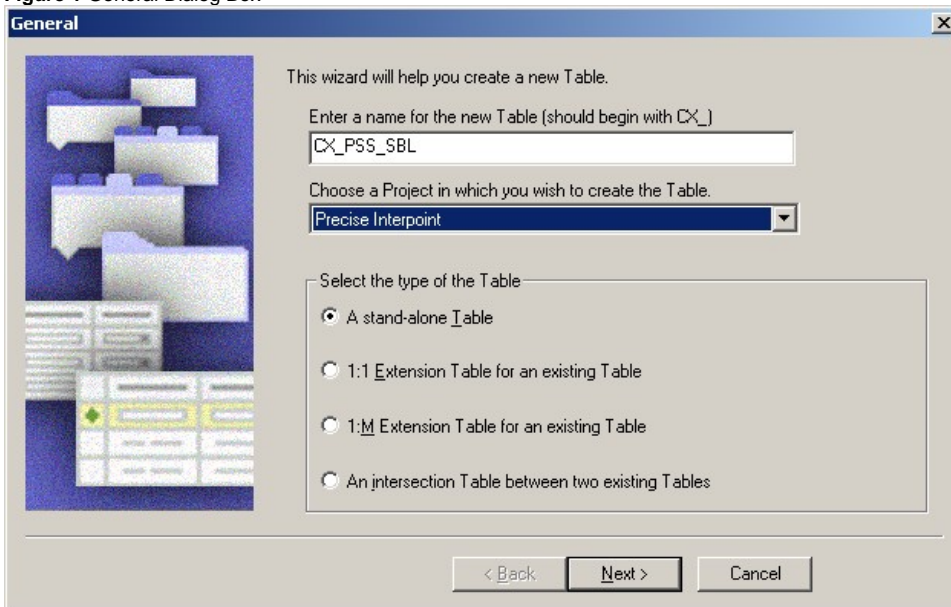
4. In the Enter a name for the new Table field, type **CX_PSS_SBL**.
5. From the Choose a Project in which you wish to create the Table drop-down list, select **Precise Interpoint**.
6. Under Select the type of the Table, select **A stand-alone Table**.
7. Click **Next**.
8. Click **Finish** to close the dialog box.
9. For versions prior to version 8.0, in the Tables workspace, click **Apply**. For version 8.0 and higher, in the Tables workspace, click **Apply/DDL**. In the displayed dialog box, click **Apply**.

10. The Apply Schema dialog box is displayed.

Figure 2 Apply Schema dialog box

11. If needed, type a valid privileged user ID and privileged user password into the relevant text boxes.

12. Click **Apply**. The dialog box closes and the table is created.

13. In the Tables panel, click **Activate** to activate the new table.

14. In Siebel Tools, on the Types tab in the Object Explorer, select **Table**. The Tables panel opens.

15. Select table **CX_PSS_SBL**.

16. On the Types tab in the Object Explorer, Expand Tables and select **Column**. The Columns panel opens.

17. From the Edit menu, choose **New Record**. A new record is added to the table in the Columns panel.

18. Add two new columns to the table as follows:

Table 3 Table with new columns

Name	User Name	Physical Type	Length
CLIENT_INFO	Client Info	Varchar	64
ACTION	Action	Varchar	32

19. For versions prior to version 8.0, in the Tables workspace, click **Apply**. For version 8.0 and higher, in the Tables workspace, click **Apply/DDL**. In the displayed dialog box, click **Apply**. The Apply Schema dialog box is displayed (see Figure 1-2).

20. If needed, type a valid privileged user ID and privileged user password into the relevant text boxes.

21. Click **Apply**. The dialog box closes and the records are created.

22. In the Tables panel, click **Activate** to activate the table with the new columns.
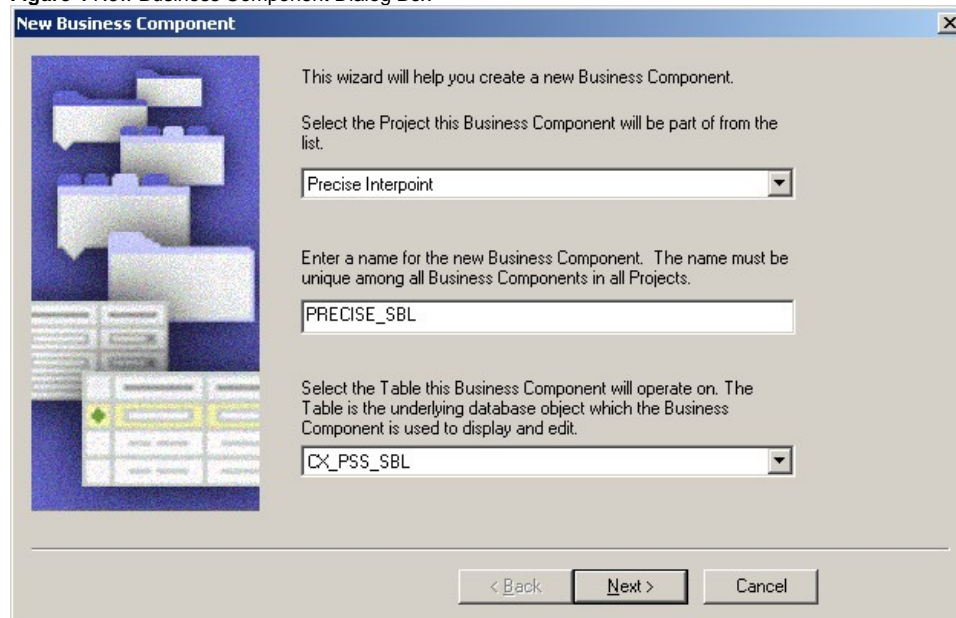
Figure 3 Siebel Repository - Column List Dialog Box
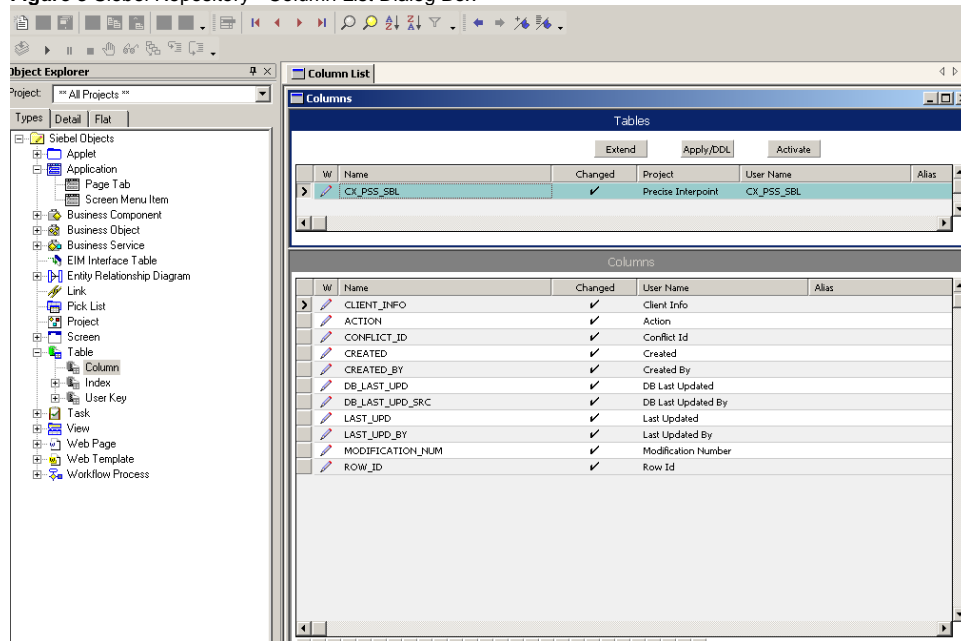
Creating the business component (Oracle)

1. In Siebel Tools, from the File menu, choose **New Object**. The New Object Wizard opens.
2. On the General tab, select **BusComp**.
3. Click **OK**. The New Business Component dialog box opens.
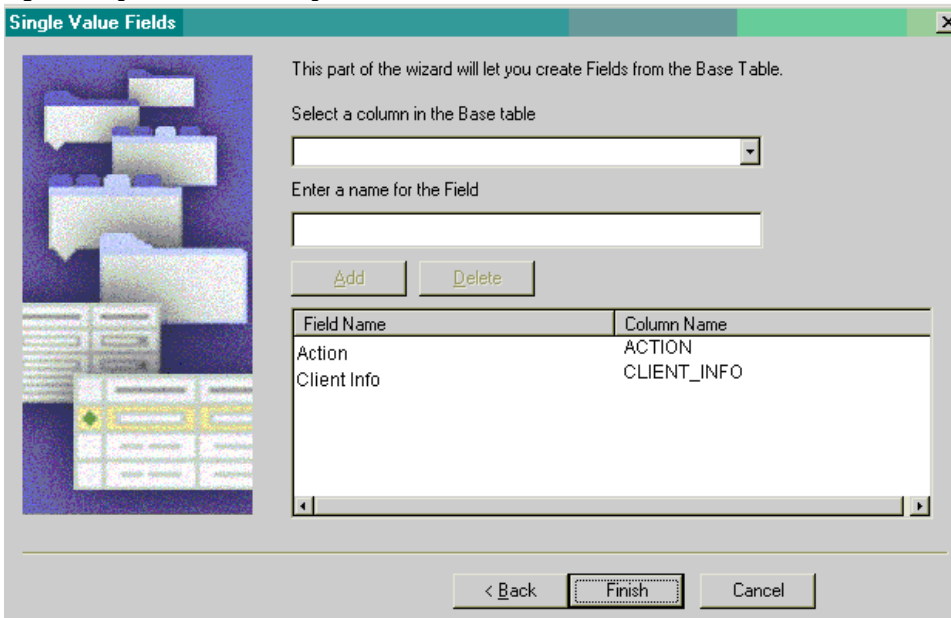
Figure 4 New Business Component Dialog Box

4. In the Enter a name for the new Business Component field, type **PRECISE_SBL**.
5. From the Select the Project this Business Component will be part of drop-down list, select **Precise Interpoint**.
6. From the Select the Table this Business Component will operate on drop-down list, select **CX_PSS_SBL**.

7. Click **Next**. The Single Value Fields dialog box opens.
   Figure 5 Single Value Fields Dialog Box

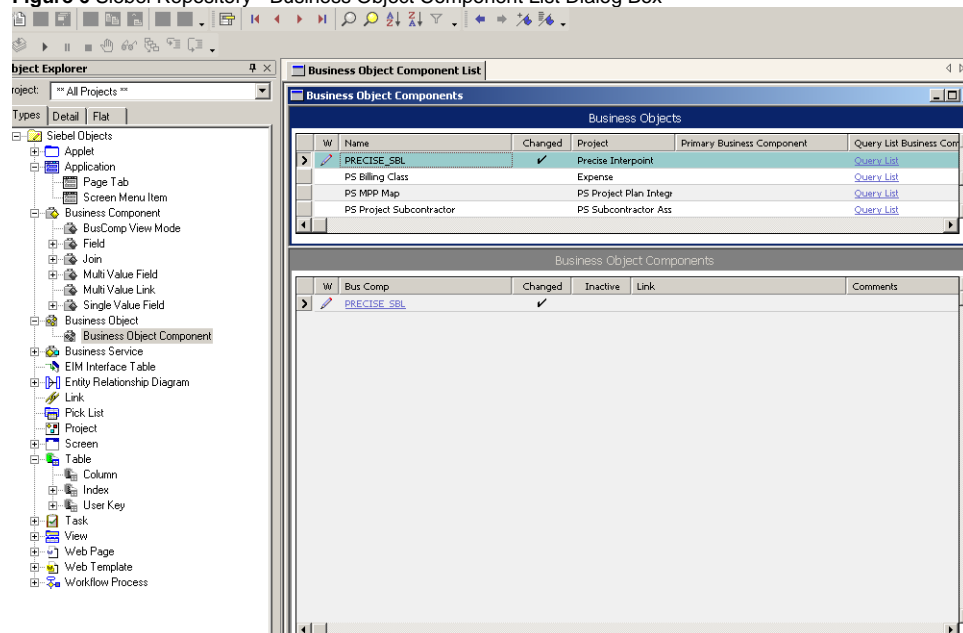8. From the Select a column in the Base table drop-down list, select **ACTION**. A default field name automatically appears in the Enter a name for the Field text box.
9. Click **Add** to move the selected row to the table.
10. From the Select a column in the Base table drop-down list, select **CLIENT_INFO**. A default field name automatically appears in the Enter a name for the Field text box.
11. Click **Add** to move the selected row to the table.
12. Click **Finish** to create the Business Component.

Creating the business object (Oracle)

1. In Siebel Tools, on the Types tab in the Object Explorer, select **Business Object**. The Business Objects panel opens.
2. From the Edit menu, choose **New Record**. A new record is added to the table in the Business Objects panel.
3. In the Name column, type **PRECISE_SBL**.
4. In the Project column, assign the new record to the Precise Interpoint project.
5. On the Types tab in the Object Explorer, select Business Object > Business Object Component.
   **Figure 6** Siebel Repository - Business Object Component List Dialog Box

6. From the Edit menu, choose **New Record**. A new record is added to the table in the Business Object Components Panel.
7. In the Bus Comp column, type **PRECISE_SBL**.
8. From the Tools menu, choose **Compile Project**.

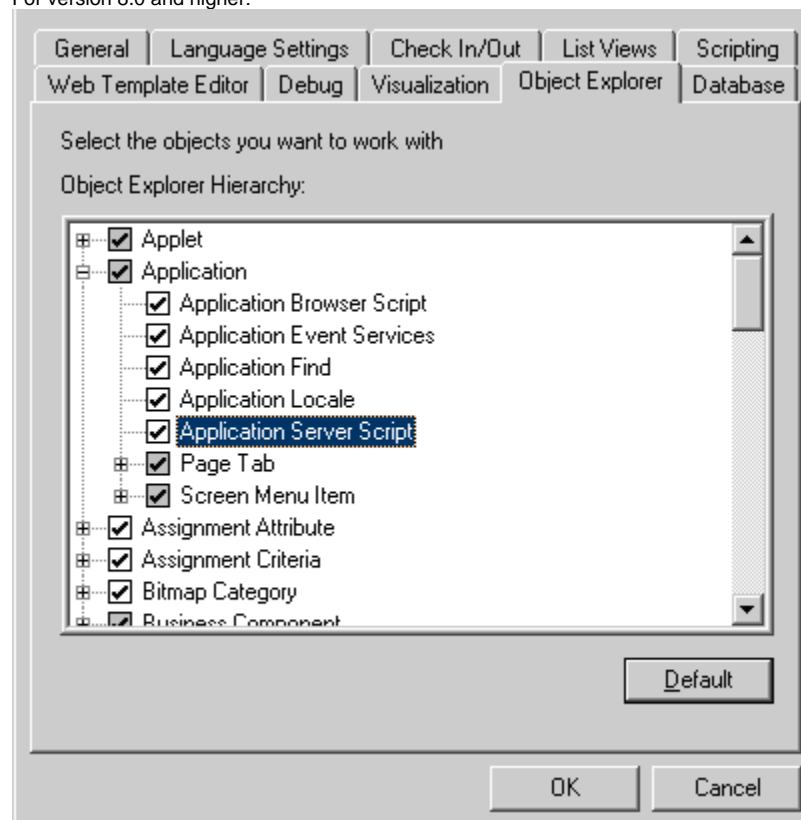## Adding the Application_PreNavigate() function (Oracle)

1. In Siebel Tools, from the View menu, select **Options**. On the Scripting tab, note the value of the Scripting Language list. If no language is selected, select the **eScript** option.
   Figure 7 Development Tools Options Dialog Box
   For versions prior to version 8.0:
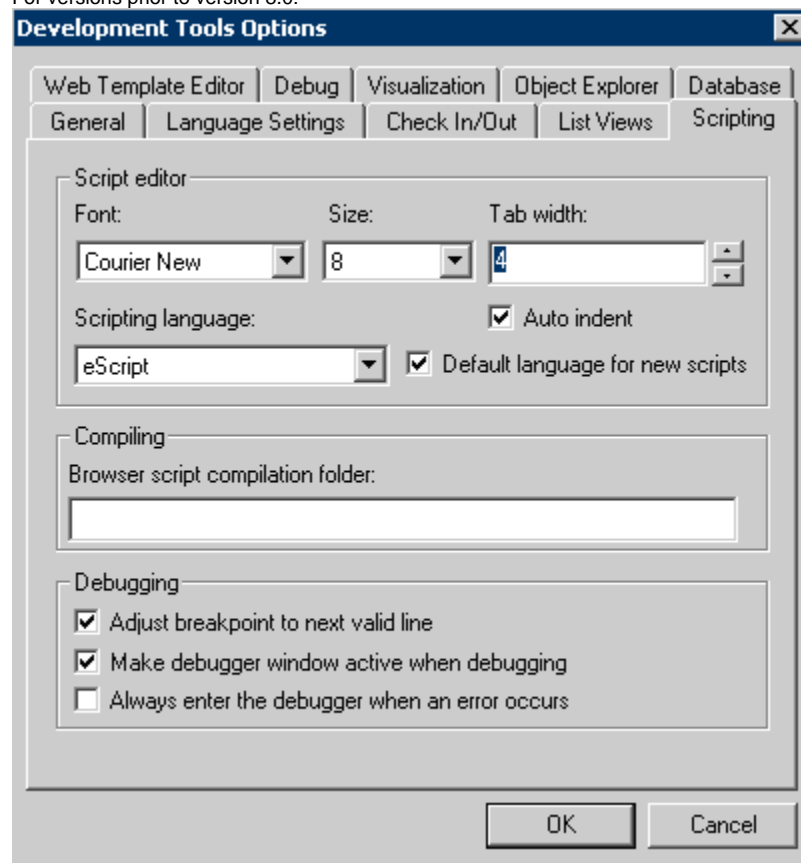
For version 8.0 and higher:

2. In the Object Explorer, on the Types tab, expand **Application**.

3. Right-click on the application to be modified and select **Lock Object**.
4. Right-click on the application to be modified and select **Edit Server Scripts**.
5. In the Server Script Editor, select the **Application_PreNavigate** function. The function's default script appears in the text panel.
   Figure 8 Siebel Repository - Application [Siebel Sales Enterprise] - Script Dialog Box
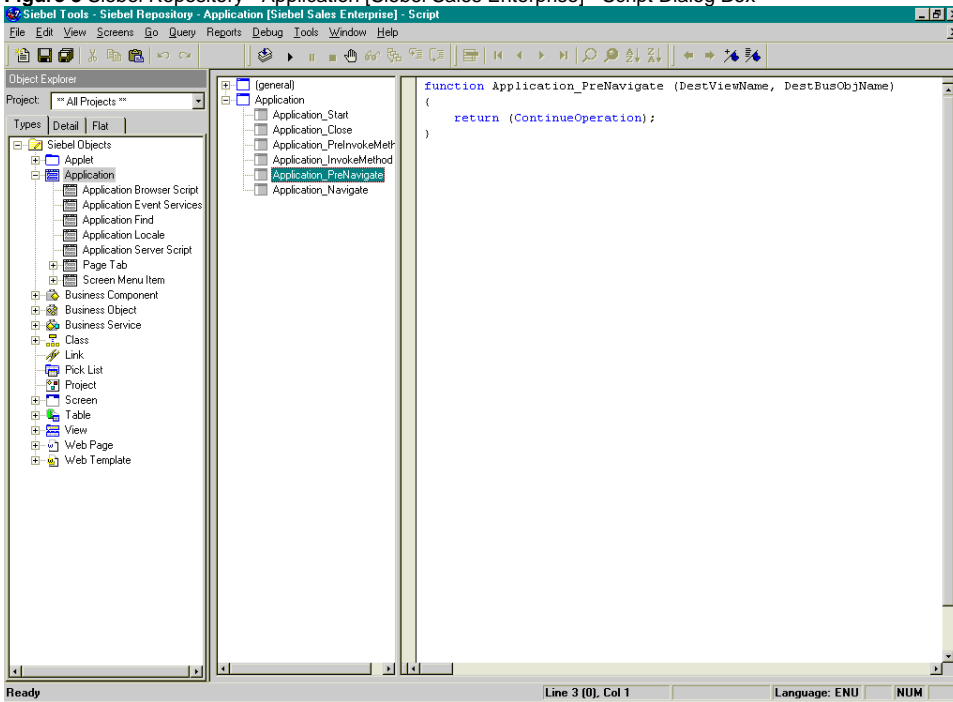
> (i) If this function already contains a custom script, do not continue. Contact Precise Technical Support for alternative instructions.

6. If the used Scripting Language is eScript, use the following script:

> (i) Modify the appName variable to match the name of the Siebel application.

```
Function Application_PreNavigate (DestViewName, DestBusObjName)
{
    try
    {
        var    oActionBO;
        var    pssBC;
        var    appName = "Siebel Sales Enterprise";
        var    action; //(32)
        var    client_info; //(64)

    oActionBO =
        TheApplication().GetBusObject("PRECISE_SBL");
            pssBC = oActionBO.GetBusComp("PRECISE_SBL");

            action = DestViewName.substring(0,32);
        client_info="S," +
        TheApplication().LoginName().substring(0,14) + "," +
            DestBusObjName.substring(0,31) + "," +
            appName.substring(0,17);

            pssBC.NewRecord(1);
        pssBC.SetFieldValue("Action", action);
            pssBC.SetFieldValue("Client Info", client_info);
            pssBC.WriteRecord();

            pssBC = null;
            oActionBO = null;
    }
    catch (obj)
    {
        pssBC = null;
        oActionBO = null;
        var sText = obj.errText;
```

```
            var nCode = obj.errCode;
        }
        finally
        {
            return (ContinueOperation);
        }
    }
}
```

7. If the used Scripting Language is Visual Basic, use the following script:

> ⓘ Modify the appName variable to match the name of the Siebel application.

```
Function Application_PreNavigate (DestViewName As String, DestBusObjName As String) As Integer
    Dim    oActionBO As BusObject
    Dim    pssBC As BusComp
    Dim    action As String
    Dim    client_info As String
    Const appName = "Siebel Field Service"

    On Error Resume Next
    Set oActionBO = TheApplication.GetBusObject("PRECISE_SBL")
    Set pssBC = oActionBO.GetBusComp("PRECISE_SBL")

    action = Left(DestViewName,32)
    client_info="S," + Left(LoginName,14) + "," + Left(DestBusObjName,31) + "," + Left(appName,17)

    pssBC.NewRecord 1
    pssBC.SetFieldValue "Action", action
    pssBC.SetFieldValue "Client Info", client_info
    pssBC.WriteRecord
    Set oActionBO = Nothing
    Set pssBC = Nothing

    Application_PreNavigate = ContinueOperation
End Function
```

8. Compile the project.

Replacing the table with a global temporary table (Oracle)

1. Log in to your Siebel database, using SQLPLUS. Make sure to log in with a privileged account.
2. Run the following statement:
   `create table CX_PSS_SBL_TAB as select * from CX_PSS_SBL where 1=2;`
3. Drop the following table:
   `DROP TABLE CX_PSS_SBL;`
4. Set the following grant for the table:
   `GRANT SELECT, INSERT, UPDATE, DELETE ON CX_PSS_SBL_TAB to SSE_ROLE;`
5. Create the following view:
   `create view CX_PSS_SBL as`
   `Select * from CX_PSS_SBL_TAB;`
6. Set the following grant for the view:
   `GRANT SELECT, INSERT, UPDATE, DELETE ON CX_PSS_SBL TO SSE_ROLE;`
7. Create the following trigger:
   `CREATE OR REPLACE TRIGGER CX_PSS_SBL_TRG`
   `INSTEAD OF INSERT on CX_PSS_SBL`
   `BEGIN`
   `DBMS_APPLICATION_INFO.SET_CLIENT_INFO (:new.client_info); DBMS_APPLICATION_INFO.SET_ACTION (:new.action);`
   `END;`
   `/`

Completing the modification (Oracle)

1. Save all changes.
2. Compile the Siebel project. This will create a new Siebel.srf file.
3. Test the project on your development machine.
4. On each of your production Siebel Servers:
   a. Stop the Siebel Server.
   b. Deploy the new Siebel.srf file
   c. Verify that the EnableScripting option in the application CFG file is set to TRUE.
   d. Start the Server.

Debugging the script (Oracle)

1. Before attempting to debug, select **Options** from the View menu.
2. Click the Debug file tab.

3. Note the file specified in the CFG file field and open it in a text editor.
4. Make sure the RepositoryFile and ApplicationName options are set correctly.
5. Make sure the EnableScripting option is set to TRUE.