PL/SQL Developer
Version Control Plug-In 1.2
User's Guide

Introduction

This document describes the PL/SQL Developer integration Plug-In for Version Control Systems. This Plug-In is the interface between PL/SQL Developer and any Version Control System that supports the Microsoft SCC Interface Specification. The Plug-In itself does not have any built-in Version Control capabilities, it relies solely on the Version Control System that you are using.

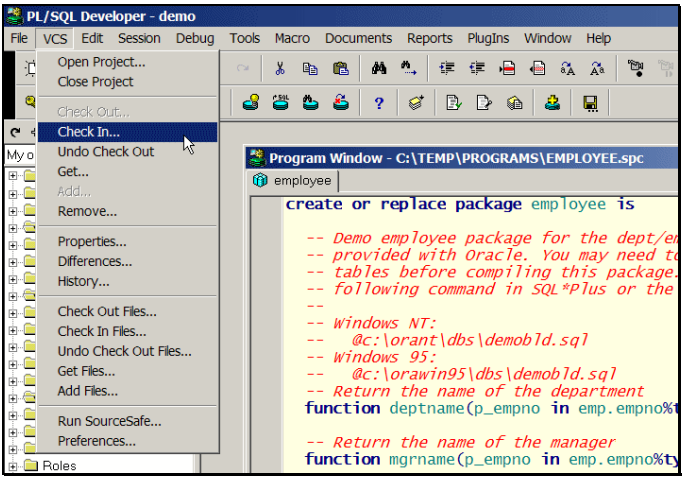
Benefits

The benefits of the VCS Plug-In come from the tight integration between the PL/SQL Developer IDE and the Version Control System:

- **Improve productivity** - You can quickly perform any VCS operation on the currently opened file in the PL/SQL Developer IDE, like check out, check in, get, add, view differences, and so on. Operations on multiple files will immediately be reflected in PL/SQL Developer's IDE.
- **Minimize errors** – Because you can work from one single IDE, the chance is minimized that you operate on the wrong file or that you forget a file.

Integration

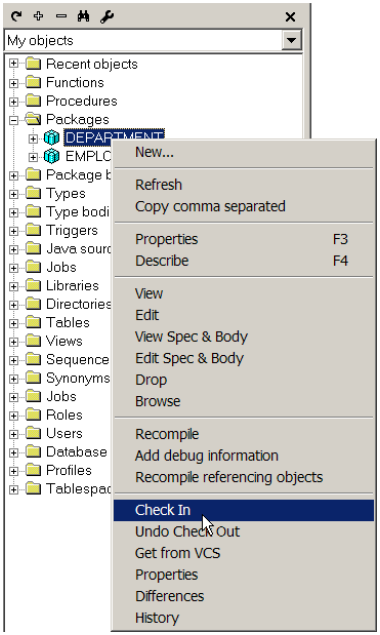
When the VCS Plug-In is installed, you will see a new *VCS* item in the main menu. This menu item contains all VCS related functions:



The Program Window, SQL Window, Test Window and Command Window also have several new VCS related items in their context menu. You can check in, check out, undo check out, get, add and remove the corresponding file, if appropriate.

Furthermore you can use the *Preferences* item of the *Tools* menu to add buttons to the toolbar for frequently used VCS functions.

If you have enabled the *Allow version control over DB objects* preference (see chapter 2.10), you will additionally see corresponding popup menu items when right-clicking on a database object in the Object Browser:



This allows you to work directly with the source of a database object in the Oracle Dictionary instead of a source file on the file system. See chapter 2.9 for more information.

1. Installation and configuration

To install the VCS Plug-In, simply run the supplied setup.exe. This will copy the Plug-In DLL (vcs.dll) into PL/SQL Developer's Plugins directory. Make sure that you have write access to this directory when you run the setup program.

After installing the Plug-In, you have a new item in the PL/SQL Developer main menu: VCS. This menu contains all items relevant to Version Control. For those functions that you will use frequently, you can assign a hotkey in the Key configuration page of the Preferences.

If the VCS menu item is disabled, this indicates that no Version Control System is detected on your PC that supports the Microsoft SCC Interface Specification. Note that this interface often needs to be installed as a separate option. For example, PVCS has a *SCC Interface* option, and MKS Source Integrity has a *32-Bit SCC Extensions* option. Make sure that you have installed the appropriate interface option for your Version Control System if the VCS menu item is disabled.

System requirements
PL/SQL Developer 2 or later (5.1.4 recommended)
A Version Control System that supports the Microsoft SCC Interface Specification

Note that version 5.1.4 or later is required if you want to check out database objects directly from the Object Browser. Earlier PL/SQL Developer versions require that you work with files on the file system.

Important PL/SQL Developer preferences
The following PL/SQL Developer preferences can be set to enforce a strict VCS approach. Both can be found on the *Options* tab page of the *Tools > Preferences* function:

- **Allow editing of database source** – Disable this preference to ensure that only work files can be edited. A work file will typically be read-only (or absent) if it is not checked out, so that it is ensured that you can only edit files that are actually checked out by you.
- **Allow compilation of read-only source files** – Disable this preference to make sure that you cannot compile a read-only work file. This ensures that you cannot accidentally compile a file that is not checked out by you, which could potentially undo the work of a colleague working on the same project if you are using the same database schema.
- **Ask to save edited database source** – Disable this if you enable the *Allow version control over DB objects* preference (see chapter 2.9), and want to work directly with the source in the Oracle Dictionary. When enabled, PL/SQL Developer will ask you to save the source to a file after each modification.

Furthermore you may want to align the PL/SQL Developer Browser *Double-click action* preference with the VCS preference *Add Spec and Body as a single file* (see chapter 2.10). If you treat the spec and body as one file in your version control system, you probably also want to view and edit them as one when double-clicking on a package or type in the Object Browser.

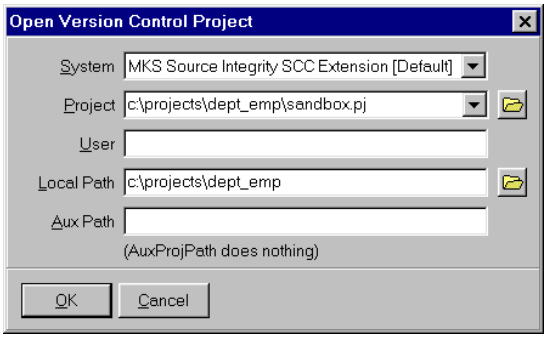
2. *Using the VCS functions*

This chapter describes all VCS function that can be found in the VCS main menu.

2.1 Opening a VCS Project

Before you can perform any VCS operation on a file, you need to open a VCS Project. As long as you have not opened a project, all corresponding menu items will be disabled.

To open a project, select the *Open Project* item from the *VCS* menu. This will display the following dialog box:



The *System* list allows you to select the Version Control System of the project. This is only relevant if you are using more than one system on your PC. It is possible to have different projects in different systems. As you switch between projects, you will also automatically switch between Version Control Systems. Note that you can set several preferences for each system, as described in chapter 2.10.

In the *Project* list you can enter the name of the project. For some Version Control Systems (like PVCS or Microsoft SourceSafe) this will be a logical name, for others (like MKS Source Integrity) this will be the full path to a project file. The list contains a history of previously opened projects. Selecting an item from this list will enter the appropriate information in the other fields of the Open Project dialog box, so that it can quickly be reopened.

The *Browse* button on the right side of the list will display a system specific *Select Project* dialog. If you select a project this way, all other fields will automatically be filled in. Not all systems support this feature. For example, MKS Source Integrity will display a *Create Sandbox* dialog, which forces you to create a new work area. In this situation you need to know the necessary information to open a project.

In the *User* field you can specify the username by which you want to login to the Version Control Project. You can usually leave this field empty.

The *Local Path* field specifies the base directory where the Version Control System should place files that are retrieved from the project, or where files can be found that are added or checked in to the project. Some Version Control Systems use the term *Work Area* for such a directory. Note that this information is very important for a proper integration, as both PL/SQL Developer and the Version Control System assume that you edit and store your work files here.

The *Auxiliary Path* has a different meaning for different Version Control Systems. This meaning is displayed below the field. Leave it empty if its meaning is unknown to you.

2.2 Checking out files

There are 2 methods to check out a file from the Version Control Project. The first method assumes that you are viewing a file in your work area, and want to check it out. The second method assumes that you want to select one or more files from the Version Control Project and check them out.

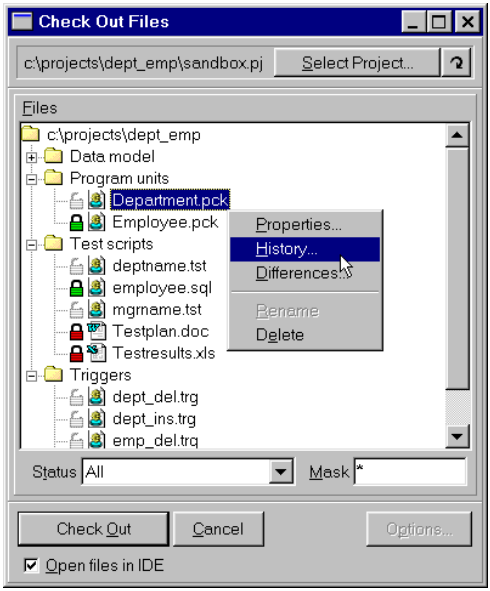
Checking out the current file

If you are viewing a source file in a window in PL/SQL Developer's IDE and wish to check it out, you can simply select the *Check Out* item from the *VCS* menu. Depending on the Version Control System and the preferences (described in chapter 2.10), this can bring up an options dialog where you can select the exact version of the file you wish to check out.

After the file is checked out, a writeable copy will exist in your work area. The contents of the corresponding window in PL/SQL Developer's IDE will be refreshed if necessary. If the file was previously read-only (which will typically be the case for a work file that is not checked out), the editor will now be modifiable.

Checking out multiple files

If you want to check out one or more source files that are perhaps not yet opened in PL/SQL Developer's IDE, select the *Check Out Files* item in the VCS menu. This will bring up the Project Files dialog that allows you to select multiple files from the project:



After selecting the appropriate files, press the *Check Out* button to check them out. The Project Files dialog has the following functions:

- **Select project** – This button allows you to select a different project, as described in chapter 2.1.
- **Refresh** – This button refreshes the contents of the file dialog. If you hold down the *Ctrl* key when pressing this button, the project will be closed and reopened.

- **Status** – Select the status of the files that will be visible in the dialog. You can view all files, all unlocked files, all files that are checked out, all files that are checked out by you, or all files that are checked out by others.
- **Mask** – Enter the filename mask of the files that will be visible in the dialog.
- **Options** – Pressing this button will bring up a *Check Out Options* dialog of the Version Control System, if supported. Some systems automatically display these options when checking out files.
- **Open files in IDE** – If this checkbox is checked, the files that are checked out will automatically be opened if PL/SQL Developer's IDE.

A lock icon visualizes the status of each file. When it is opened and gray, the file is unlocked. If it is closed and green, you have checked it out. If it is closed and red, another project member has checked it out.

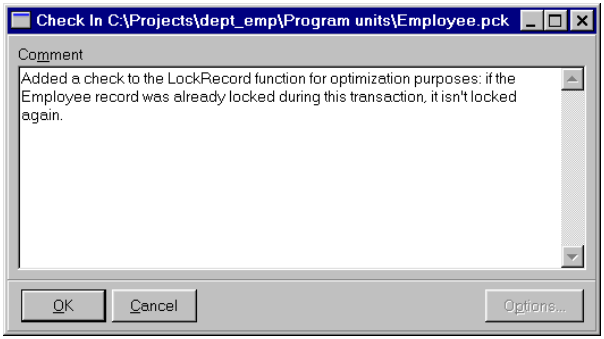
If you right click on a file, this will bring up a context menu that allows you to view the properties of the file, the differences between the file and the work file, and the history of the file. You can also rename or delete a file, if the Version Control System supports it.

2.3 Checking in files

Just like with the check out function, you can either check in the file you have currently opened in PL/SQL Developer's IDE, or you can check in multiple files from the Project Files dialog.

Checking in the current file

To check in a file that you have previously checked out, you can simply open it in PL/SQL Developer's IDE and select the *Check In* item from the *VCS* men. This menu item will only be enabled if the file was indeed checked out by you. After selecting this menu item you will be prompted for a change comment:



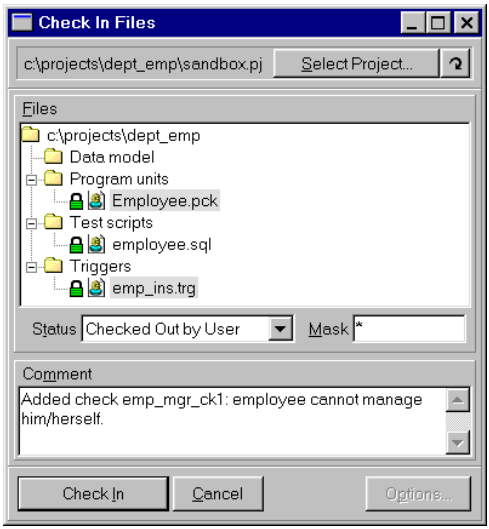
This dialog also allows you to set some *Check In Options*, if the Version Control System supports it. Some systems will provide their own dialog to specify the change comment and options. In this case you may want to disable the dialog above through the preferences described in chapter 2.10.

If the current file was changed in the IDE and not yet saved, it will be saved it to disk before it is checked in. After being checked in, you will typically be left with a read only copy of the work file. In that case the file will also be read only in the IDE.

Checking in multiple files

Very often you do not want to check in individual files, but want to check in several files at once with one change comment. You can do so by selecting the *Check In Files* item from the *VCS* menu. This will

bring up the Project Files dialog that displays all files that were checked out by you. This dialog now also has a *Comment* field where you can enter the change comment:



If any of these files is opened and changed in PL/SQL Developer's IDE and not yet saved, it will be saved it to disk before they are checked in. After being checked in, you will typically be left with a read only copy of these work files. Any file that was open in the IDE will now also be read only.

2.4 Undoing a check out

If you have checked out a file and come to the conclusion that this was not necessary, you can undo the check out by selecting the *Undo Check Out* or *Undo Check Out Files* item from the *VCS* menu. The *Undo Check Out* item operates on the current file in PL/SQL Developer's IDE. The *Undo Check Out Files* item allows you to select multiple files from the Project File dialog.

Any changes that result from this action will immediately be reflected in PL/SQL Developer's IDE. For example, the original version of the file may be restored in the work area, and the file may become read only.

2.5 Getting files

To get a specific version of a file into your work area and into PL/SQL Developer's IDE, you can use the *Get* or *Get Files* item in the *VCS* menu.

The *Get* item will get a version of the file you have currently opened in PL/SQL Developer's IDE. If the Version Control System supports it, and if the preferences are set accordingly (as described in chapter 2.10), you will be prompted to specify some options. You may be able to specify the exact file version, whether the copy is read only, and so on.

The *Get Files* item allows you to select multiple files from the Project File dialog, which can automatically be opened in PL/SQL Developer's IDE. You can specify the options in this dialog as well.

2.6 Adding files

You can add files to the Version Control Project by selecting the *Add* or *Add Files* item from the *VCS* menu. The *Add* item adds the file you have currently opened in PL/SQL Developer's IDE to the Version Control Project. The *Add Files* item will bring up an *Add Files to Project* dialog that allows you to select multiple files from your work area.

Just like with the check in function, the file will be saved before adding it if necessary, and you can specify a comment and options if the Version Control System supports it and if the preferences are set accordingly. Any changes that result from this action will immediately be visible in PL/SQL Developer's IDE.

2.7 Displaying file information

You can display information from the Version Control Project about the file you have currently opened in PL/SQL Developer's IDE by selecting the *Properties*, *Differences*, and *History* item from the VCS menu.

The *Properties* that are displayed depend on the Version Control System that you are using. It will usually display the last version, its change comment, author, creation date, status and so on.

The *Differences* function will display the differences between the work file and the file in the Version Control Project. It may be necessary to save the file if you have made changes to it, otherwise you are comparing a previously saved version.

The *History* of the file will usually display a list of properties for each version of the file in the Version Control Project.

2.8 Running the Version Control System

To run the Version Control System with the project that you have currently opened, simply select the *Run…* item from the *VCS* menu.

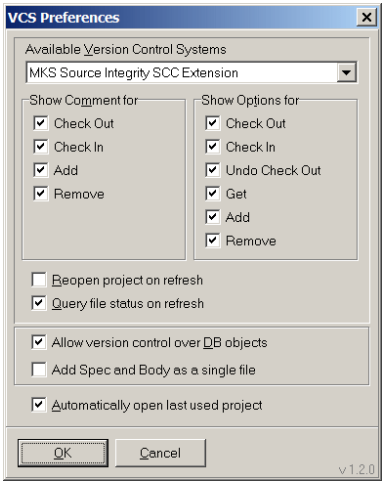
2.9 Working directly with database objects

If you do not use source files on the file system in your development process, but instead work directly with the sources in the Oracle Dictionary, you may want to use your VCS in the same way. Instead of checking out a file, you check out a database object. To enable this, you must first set the *Allow version control over DB objects* preference (see chapter 2.10). After doing so, you will see the VCS functions in the popup menu when right-clicking on a function, procedure, package, type, trigger or view object.

When using a VCS function on a database object, you will indirectly be using a work file that is stored in the Local Path (Work Area) of the VCS project. For example, if the Local Path of your project is u:\projects\deptemp, and you check out the employee package body, a work file with the name u:\projects\deptemp\employee.bdy will be created by the VCS. The actual extension of the work file depends on the *File extensions* preferences in PL/SQL Developer.

2.10 Preferences

You can specify several preferences that affect the behavior of the VCS Plug-In:



These preferences have the following effect:

- **Automatically open last used project** – When you start PL/SQL Developer, the Version Control Project that was opened the last time will automatically be opened again.
- **Allow version control over DB objects** – When enabled, you can use the VCS functions directly on a database object in the Object Browser. See chapter 2.9 for more details.
- **Add Spec and Body as a single file** – If the *Allow version control over DB objects* preference is enabled, this preference controls how a package or type is added. When enabled, a package or type will be added as one file with both the specification and body. When disabled, the specification and body must be added as separate files.

The remaining preferences can be defined separately for each Version Control System. Use the list to select the Version Control System for which you want to change or view the preferences.

- **Show Comment for** – When the corresponding operation is performed, PL/SQL Developer will explicitly allow you to enter a change comment. If the Version Control System already prompts you for a change comment for such an operation, you should uncheck the preference.
- **Show Options for** – When the corresponding operation is performed, PL/SQL Developer will explicitly allow you to set the options for this operation. If the Version Control System already prompts you for the options for such an operation, you should uncheck this preference.
- **Reopen project on refresh** – If the preference is checked, the project will be closed and reopened if you press the *Refresh* button in the Project File dialog. Some Version Control Systems do not immediately see changes if a project is not reopened.
- **Query file status on refresh** – When a file list is retrieved from the Version Control Project, the file status (unlocked, checked out) of each individual file may not always be correct. Check this preference to explicitly query this information. It may lead to some performance degradation when opening or refreshing a project. MKS Source Integrity is an example of a Version Control System that requires this preference.