



Pagos Spreadsheet Component for .NET 5

User Guide

December 2009

Pagos, Inc.
1031 Farmington Avenue
Farmington, CT 06032
USA
Tel: (860) 674 – 9100
Fax: (860) 674 – 8430
Email: info@pagos.com

© 2006–2009 Pagos, Inc. All rights reserved.

The information contained in this document represents the current view of Pagos, Inc. on the issues discussed as of the date of publication. Because Pagos must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Pagos, and Pagos cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. **PAGOS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.**

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Pagos Inc.

Microsoft and Microsoft Excel are trademarks of Microsoft Corporation in the United States, other countries, or both.

Table of Contents

1.1	Introduction	4
1.2	What's new?	4
1.2.1	Version 5.1	4
1.2.2	Version 5.0	5
1.2.3	Version 4.6	5
1.2.4	Version 4.5	5
1.2.5	Version 4.4	6
1.2.6	Version 4.3	6
1.2.7	Version 4.2	6
1.2.8	Version 4.1	7
1.2.9	Version 4.0	7
2.1	Enterprise Edition	8
2.2	Standard Edition	8
2.3	Lite Edition	8
3.1	System Requirements	8
3.2	Side by side installation with PSC 3.....	9
3.3	Installation with setup wizard.....	9
3.4	Upgrading from PSC 4.x to 5.0	9
3.5	Installation without setup wizard.....	10
5.1	Configuration file	14
5.2	Configuration data	14
5.3	Location	15
6.1	Introduction	15
6.2	Migration and Debugging with ssdiff.....	16

1. Overview

1.1 Introduction

Pagos Spreadsheet Component 5 (PSC) is a Microsoft Excel-compatible engine which can be embedded into web applications to process Excel files on a server. This simplifies maintaining and deploying Excel models as new versions don't need to be sent to end users but can be installed, e.g., on a company's intranet server. Furthermore, maintenance and license costs can be decreased, as end users don't need to have Excel versions installed any longer.

PSC is available as Lite, Standard and Enterprise Edition on Windows. All editions provide a COM, a .NET and a Java interface to integrate PSC easily on today's platforms. PSC is also available on various Unix platforms. On Unix, only the Enterprise Edition with the Java interface is available.

All editions provide very similar interfaces. They consist of a few objects used primarily to load an Excel file and access worksheet cells. It is possible to change values, formulas and cell formats in a worksheet or to fetch values, formulas and cell formats from a worksheet. Whenever a value is fetched, the formula of the respective cell is processed by the engine to update the value (if, for example, values in depending cells have been changed). The engine can be used in popular server-side development environments such as ASP.NET and JSP. Web applications built with server-side scripts enable end users to use Excel models in their web browser for both calculating and reporting.

In order to support developers PSC ships a migration and testing utility called Ssdiff.¹ Ssdiff helps to migrate Excel-based applications to PSC. It can examine Excel files, measure loading, and processing times, measure memory consumption, and can check whether formula results in PSC match Excel results.

Although PSC has been developed for server-side usage, it is possible to include the engine in a stand-alone desktop application. The desktop application can be shipped and installed by end users who can then use PSC internally.

However, every PSC installation must be activated to use the engine. For trial purposes the Lite Edition can be used for 30 days before it must be activated.

An activation wizard is used to activate PSC. If it is to be used in a desktop application, it is recommended that the setup file be shipped separately and end users asked to install PSC first. End users must then buy a license to activate before they can use the desktop application to use PSC internally.

1.2 What's new?

1.2.1 Version 5.1

The latest version 5.1 was released on December 7, 2009:

¹ Only Enterprise Edition.

- Image support
- Merged cell support
- Support for relative references
- Cut/Copy/Paste support
- Support for Fill Down
- Chart improvements
- Improvements in memory consumption
- Performance improvements
- Improved number formatting

1.2.2 Version 5.0

The latest version 5.0 was released on April 13, 2009:

- Reduced number of DLLs
- Support for Visual Studio 2008
- Pivot Table improvements
- Support for 5 new worksheet formulas: DCOUNTA, ODDFPRICE, ODDFYIELD, ODDLPRICE, ODDLYIELD
- Chart Improvements

1.2.3 Version 4.6

The latest version 4.6 was released on September 29, 2008:

- Support for reading and writing Excel 2007 (XLSX) file format
- Support for editing merged cells
- Misc improvements in the Chart Module

1.2.4 Version 4.5

The latest version 4.5 was released on February 15, 2008:

- Chart support is improved. Following chart types are added:
 - Stock: High-Low-Close
 - Stock: Open-High-Low-Close

- Bubble
- Support for Text Property
- Support for Goal Seek feature

1.2.5 Version 4.4

The latest version 4.4 was released on December 3, 2007:

- Pivot tables in Excel files are loaded and can be accessed and manipulated through the API.
- 25 Excel formulas were added.
- International Excel files can be loaded and saved correctly as PSC 4.4 supports Unicode.
- The .NET edition is based on the .NET 2.0 Framework.

1.2.6 Version 4.3

PSC 4.3 was released on June 29, 2007:

- The engine was optimized for multithreaded applications: Workbooks can now be accessed concurrently by different threads without threads having to wait for each other.
- Built-in chart engine: Charts contained in XLS files are parsed and can be accessed and saved to files.
- More than 40 formulas have been improved and made more Excel-compliant.

1.2.7 Version 4.2

PSC 4.2 was released on April 6, 2007:

- The engine went through a major overhaul for a better performance and more flexibility. It supports now dynamic worksheets which grow when cells are inserted and shrink when cells are deleted. PSC now also supports setting formulas in cells dynamically and changing worksheets on the fly just as Excel.
- PSC can now read and save workbooks in the Excel 2003 XML file format.
- PSC supports cell formats which are loaded when a workbook is read from an Excel 2003 XML file and saved when a workbook is stored as an Excel 2003 XML file. Cell formats can be accessed via the API.
- Lookup formulas have been improved for a better performance.

1.2.8 Version 4.1

PSC 4.1 was released on October 18, 2006:

- It is the first version to support 300 Excel formulas.
- PSC can be used for the first time on Unix. The Java edition of PSC is now officially supported on Red Hat Enterprise Linux 4 (on Intel x86) and Solaris 10 (on Sparc).

1.2.9 Version 4.0

PSC 4.0 was released on July 7, 2006 on Windows, supporting various platforms and programming languages including COM, .NET and Java. While PSC 4.0 seems to be just another major version there have been many far reaching internal changes :

- Brand new and completely overhauled architecture based solely on the official C++ standard and Boost C++ libraries, the precursors of the next C++ standard version to come.
- Native spreadsheet engine used by all editions for faster processing than pure .NET or Java spreadsheet components.
- Two versions of each interface with one having explicit multithreading support and synchronizing threads accessing PSC automatically.²
- COM interface uses MTA for best performance, which can especially be seen on IIS 6 as it can be configured to use MTA for ASP web pages.
- Java interface without COM dependency for better performance.
- Full support for matrix formulas. New testing utility, Ssdiff, to support migrating to PSC.³

² Only Enterprise Edition.

³ Only Enterprise Edition.

2. Editions

2.1 Enterprise Edition

The Enterprise Edition has full support for everything which has been developed for PSC 4. It is the successor of the Professional Edition of PSC 3. There are no restrictions – all features and functions are available. This is the recommended edition to be used on servers.

2.2 Standard Edition

The Standard Edition can only parse XLS files with a size less than 1 MB and XML files with a size less than 6 MB. The Standard Edition does not support the external formulas SQLOPEN and SQLGET.

Interfaces are not synchronized and can not be safely used in a multithreaded application. Ssdiff is missing and is not shipped with the Standard Edition either.

2.3 Lite Edition

The Lite Edition has the same restrictions as the Standard Edition but can only parse XLS files with a size less than 200 KB and XML files with a size less than 600 KB. However the Lite Edition supports a 30 day trial period: PSC can be used for 30 days before it must be activated.

3. Installation

3.1 System Requirements

The following table lists the minimum system requirements to use PSC. Recommendations are given in parentheses:

	Developer System	Server System
Operating Systems	Windows 2000 or higher (Windows XP Pro recommended) with Windows Installer 3	Windows 2000 or higher (Windows XP Pro recommended) with Windows Installer 3
RAM	256 MB (512 MB recommended)	512 MB (1024 MB recommended)
Disk Space Requirements	10 MB	10 MB

The minimum RAM highly depends on the size of the Excel files to be loaded. The spreadsheet engine of PSC allocates less than 7 MB. Excel files, however, require much more

RAM. While small Excel files might allocate, for example, only 10 to 20 MB, huge Excel files can allocate several hundred MB.

Depending on the interface the following system requirements must be met, too:

	Developer System	Server System
PSC for COM ⁴		(IIS 6 with ASP as MTA recommended)
PSC for .NET	.NET 2.0 or higher (Visual Studio 2005 recommended)	.NET 2.0 or higher
PSC for Java	J2SDK 1.5 or higher	J2RE 1.5 or higher

3.2 Side by side installation with PSC 3

PSC 4 can be installed side by side with PSC 3. It is not necessary to uninstall PSC 3. Installing PSC 4 does not affect any programs which use PSC 3. There is no risk in installing PSC 4 if another version of PSC 3 is already installed.

3.3 Installation with setup wizard

PSC can be easily installed by following the steps in the setup wizard. Installing PSC is not much different from installing any other Windows application.

However, after PSC is installed you can not start and use a new application by clicking on an icon in the Windows start menu. PSC is not a Windows application but a component which can be used by developers, and can be integrated in other applications.

In the program group which will be created by the setup wizard you will find an application called Ssdiff.⁵ This application can be started directly. However it is only a migration tool supporting developers moving Excel files from Excel or other applications to PSC. In order to benefit from all functions of PSC you need to integrate the component in another application.

3.4 Upgrading from PSC 4.x to 5.0

If you use PSC.4.x and want to upgrade to the latest version 5.0 you should uninstall previous version of PSC. Don't install PSC.5.0 side by side with PSC.4.x.

⁴ No additional system requirements needed as COM is part of the Windows operating systems and available by default.

⁵ Only Enterprise Edition.

3.5 Installation without setup wizard

It is possible to install PSC on a computer without using the setup wizard. PSC is loosely coupled with the operating system. For example, PSC configuration data is stored in a XML file instead of the Windows registry.

Installing PSC on a computer without using the setup wizard is simply a matter of copying libraries. All DLL files which are installed by the setup wizard to the application folder must be copied to another computer. It is necessary to run the setup wizard on a computer first in order to get these DLL files, as they are not offered separately.

After the DLL files have been copied to the other computer, the interfaces you want to use to access PSC must be copied. If you want to use the COM interface of PSC it is also necessary to register the COM interface on the other computer. This requires access rights to the Windows registry – COM doesn't work without having access to the Windows registry.

After all files have been copied, the last step is to activate PSC. The 30 day trial period of the Lite Edition is only available when you use the setup wizard to install PSC. Therefore activating is absolutely required on the other computer to make use of PSC. Either use the activation wizard which also can be simply copied, or contact Pagos if you can't run a Windows application on the other computer (if it is for example a web server where you only have FTP access).

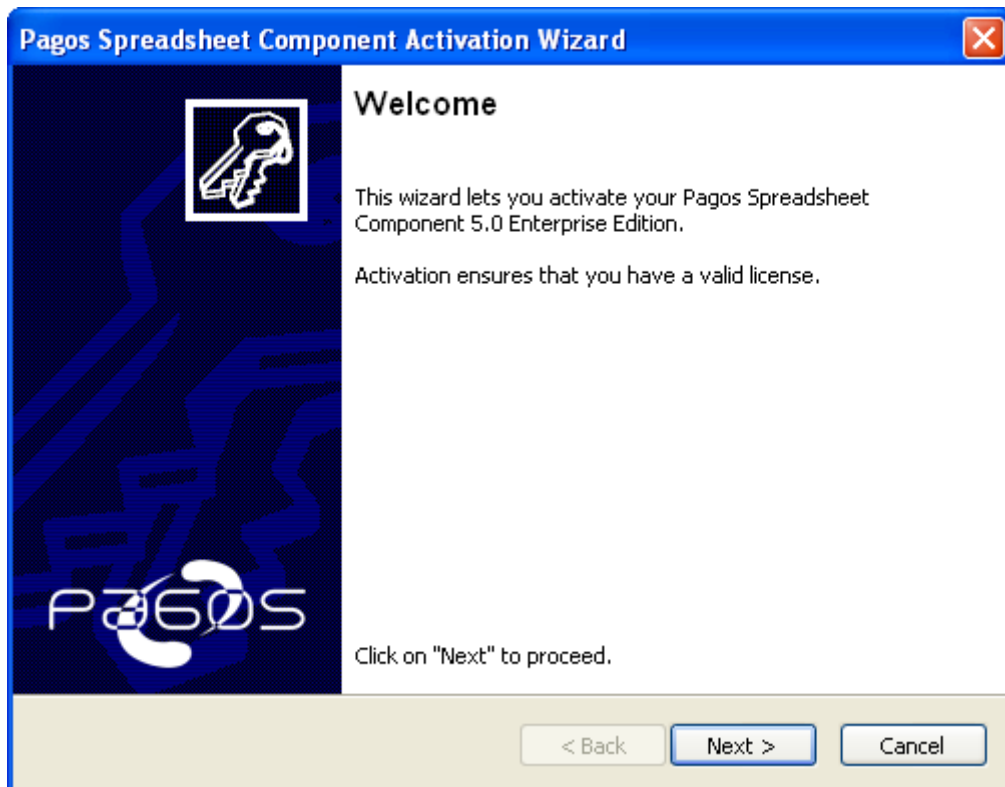
4. Activation

You can use and evaluate the Lite Edition of PSC for a trial period of 30 days. After the trial period has expired, you must activate it. The Standard and Enterprise Edition come without a trial period and must be activated immediately after installation.

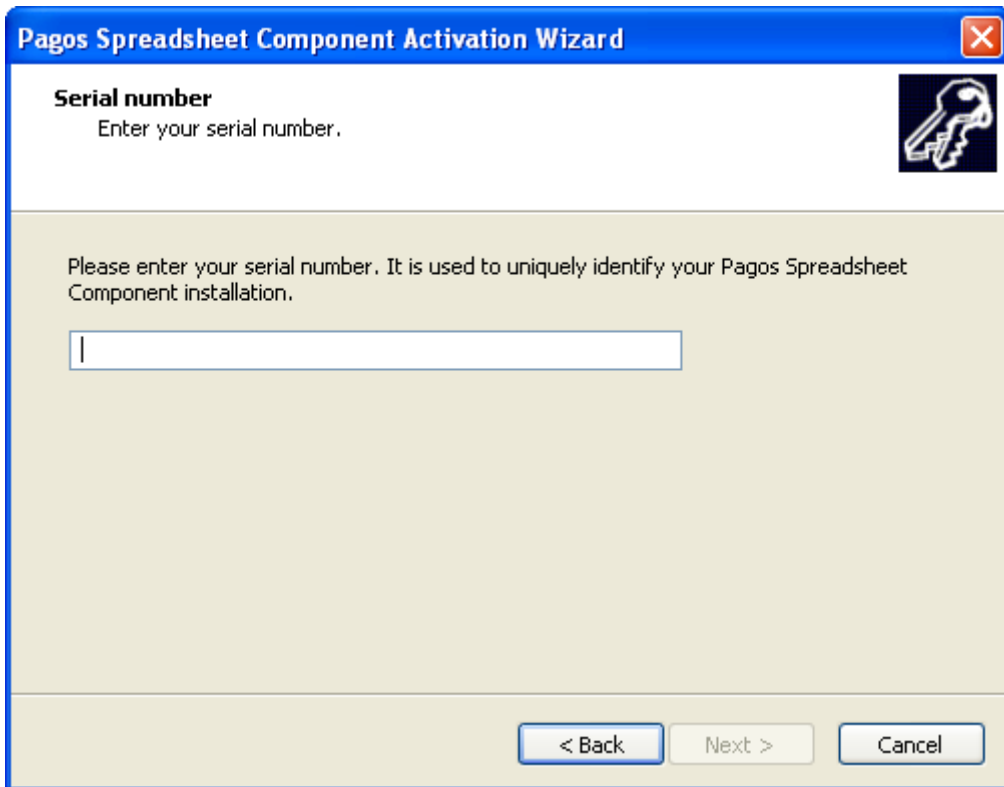
You need to buy a license to be able to activate PSC. Activation ensures that you have a paid, valid license.

In order to activate PSC you must use the activation wizard. You must use the activation wizard on the same computer you want PSC to be activated on. The license is bound to the hardware and can not be transferred from one computer to another.

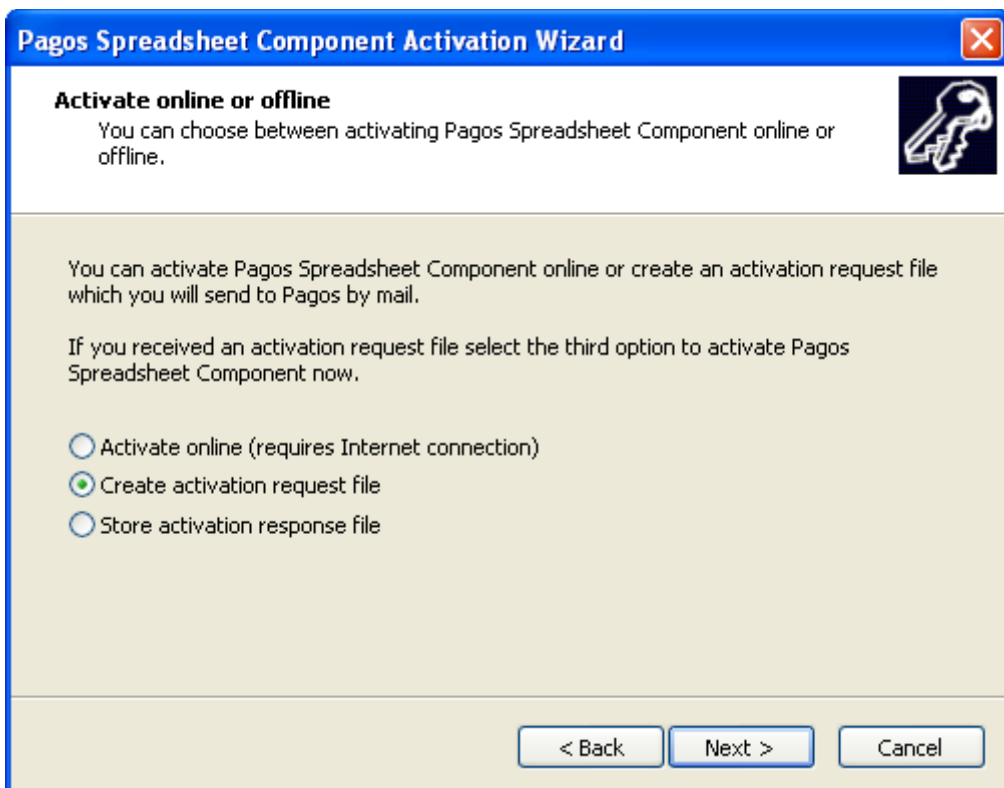
You can activate a PSC installation online or offline. The latter is done with files which must be exchanged with Pagos by mail.



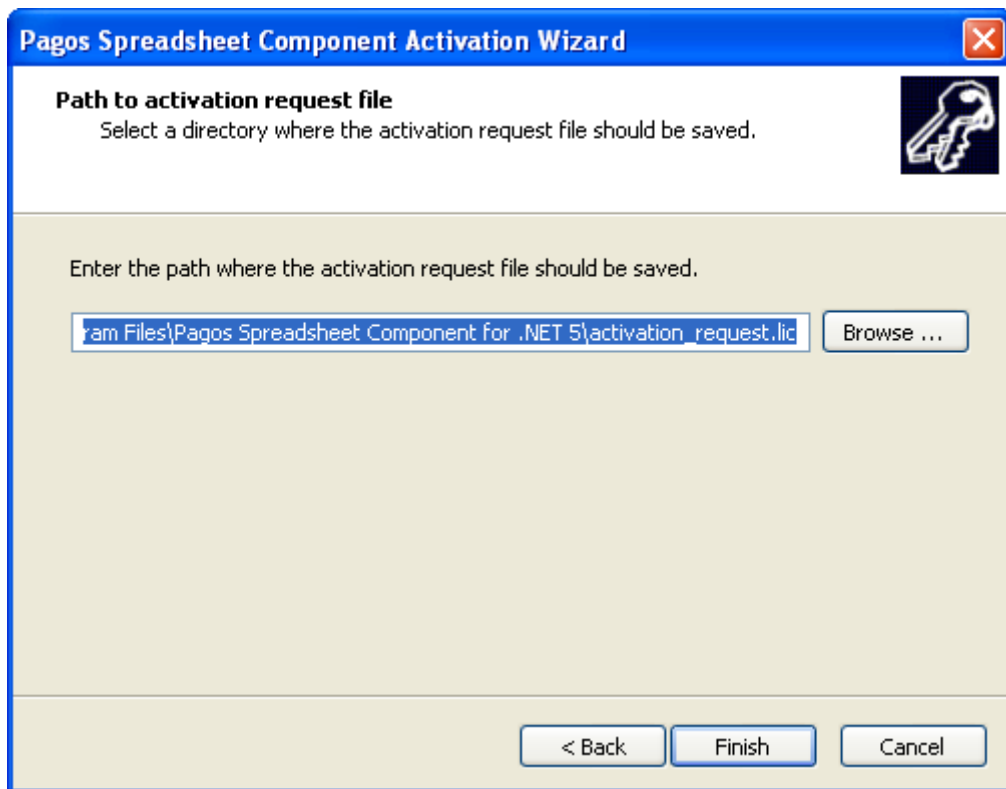
After a welcome page, the activation wizard asks for a serial number.



Serial numbers are used to identify unique PSC installations. The activation wizard checks what the user enters. Only after entering a valid serial number can the user proceed.



Activating a PSC manually is the recommended way. To do so, an activation request file has to be created and sent to Pagos. Within 48 hours an activation response file is sent back. It must be installed with the activation wizard.



Some data is necessary to activate a PSC installation. Both the online and offline activation use the same data, which is forwarded to the activation server or stored in the activation request file. Required data for activation is:

- Serial number
- Hardware ID
- Edition
- Interface
- Version

The hardware ID is a hash value and does not reveal any information about a computer system. It is impossible to learn anything about a computer system by looking at the hardware ID.

Optional data that is also forwarded but not required for activation is:

- Username
- Organization
- Number of CPUs

The username and organization can be removed from the configuration file prior to activating. Then activating a PSC installation is completely anonymous.

No other data is forwarded to activate a PSC installation – either when activating online or offline.

5. Configuration

5.1 Configuration file

The configuration of each PSC edition is based on one XML file. If you have all three PSC editions with the COM, .NET and Java interface installed you should have three configuration files.

By default, a configuration file is stored in the application directory of PSC. It is automatically created by the setup wizard and stores the username and organization which were entered by the user in the setup wizard.

The configuration files for the three PSC editions are called `psc_com.xml`, `psc_net.xml` and `psc_java.xml`. All data which is required to configure PSC is stored in these XML files. There is no data stored elsewhere – neither in the Windows registry nor in any other file.

The configuration files also store the serial number and the license when a PSC installation has been activated. The activation wizard expects to find the configuration file in the application directory. After activating PSC the activation wizard stores the serial number and the license in the configuration file.

When PSC is used by an application, it loads configuration data from the configuration file in the application directory by default. However, some applications might have no access to the application directory. For example, a web application might run in a restricted environment with access to only a very few directories.

For a more flexible configuration the configuration file can be copied to another directory. When PSC is instantiated a pathname can be specified to tell PSC which configuration file it should use. This enables applications running in a restricted environment to pass configuration data – including the license – to PSC.⁶

5.2 Configuration data

As of PSC 4.4 only four entries are allowed and used in the configuration file:

- **Username:** Stores the username which is entered by the user in the setup wizard. The username is forwarded to the activation server when PSC is activated. If you prefer an anonymous activation remove the username from the configuration file prior to activating PSC.
- **Organization:** Stores the organization which is entered by the user in the setup wizard. The organization is forwarded to the activation server when PSC is activated. If you prefer an anonymous activation remove the organization from the configuration file prior to activating PSC.

⁶ This is not supported by PSC for COM. PSC for COM requires the configuration file to be located in the application directory.

- **Serialkey:** Stores the serial key which is entered in the activation wizard when PSC is activated. Never change the serial key in the configuration file manually as the activation wizard verifies the format of serial keys.
- **License:** Stores the license which is saved by the activation wizard after PSC has been activated. Never change the license as it will make your PSC installation unusable.

Any other entry in the configuration file is ignored. There is no other data which can be added to the configuration file.

5.3 Location

When PSC is installed the configuration file is copied to the application directory. When you activate PSC the activation wizard automatically uses the configuration file in the application directory. However, after activation you are free to copy the configuration file. This might be necessary; for example, server applications that use PSC might not have access rights to read a file from the application directory of PSC.

The configuration file can be copied to any directory. When PSC is instantiated it is possible to specify a pathname to a configuration file. Thus you are flexible to put the configuration file in any directory you want.⁷

By default the configuration file must be in the same directory where psc.dll is located. If PSC is instantiated without specifying a pathname to a configuration file, psc.dll tries to open the configuration file in the same directory where it is located. For example, if you copy psc.dll to the Windows folder you must also copy the configuration file to the Windows folder, if you don't want to specify a pathname when instantiating PSC.

6. Software Integration

6.1 Introduction

Integrating PSC enables an application to process Excel files. It is then possible to implement business logic in an Excel file which can be easily changed and replaced instead of hard-coding business logic in a programming language.

The kernel of PSC – the engine – is shipped as dynamic libraries. It is implemented in C++ for best performance. While each and every application can benefit from the superb performance it is not always easy to talk to dynamic C++ libraries directly. Therefore PSC encapsulates the engine.

PSC comes with three interfaces which make it much easier to integrate the engine in an application. Instead of talking to a C++ interface in a dynamic library developers can access the ready-to-use interfaces depending on what platforms and with what programming language you work.

⁷ This is not supported by PSC for COM. PSC for COM requires the configuration file to be located in the application directory.

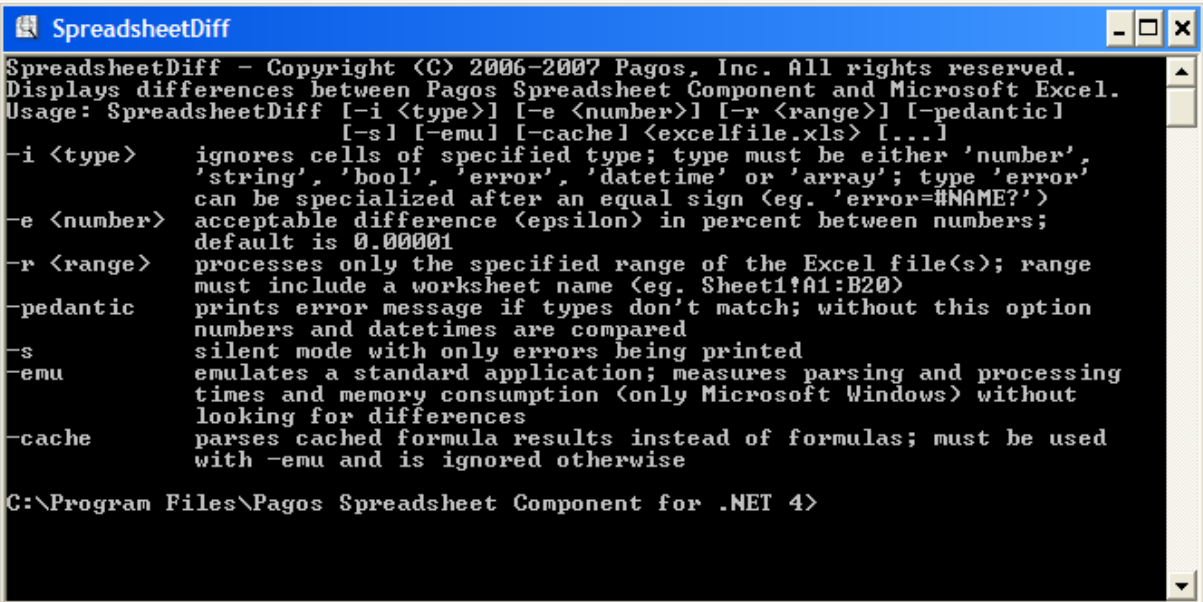
The help file shipped with PSC contains a tutorial, object model and references for every interface.

6.2 Migration and Debugging with ssdiff

Ssdiff is a console application which processes Excel files and checks if PSC calculates the same results as Excel. It replaces the test menu item in the Excel add-in which was shipped with PSC 3.x. Ssdiff is only shipped with the Enterprise Edition.

While the main purpose of ssdiff is to make sure that an Excel file is correctly processed and returns the same results in PSC as in Excel it can also be used to optimize an Excel file. Ssdiff supports an emulation mode in which it behaves like a standard application using PSC to load and process an Excel file. In emulation mode, ssdiff shows processing times and memory consumption.

Ssdiff gives immediate feedback when an Excel file is changed. It can handle any Excel file – it is not necessary to create specific test applications. Thus it is possible to see easily which changes in an Excel file boost the performance of PSC.



```
SpreadsheetDiff - Copyright (C) 2006-2007 Pagos, Inc. All rights reserved.
Displays differences between Pagos Spreadsheet Component and Microsoft Excel.
Usage: SpreadsheetDiff [-i <type>] [-e <number>] [-r <range>] [-pedantic]
                        [-s] [-emu] [-cache] <excelfile.xls> [...]
-i <type>             ignores cells of specified type; type must be either 'number',
                        'string', 'bool', 'error', 'datetime' or 'array'; type 'error'
                        can be specialized after an equal sign (eg. 'error=#NAME?')
-e <number>           acceptable difference (epsilon) in percent between numbers;
                        default is 0.00001
-r <range>            processes only the specified range of the Excel file(s); range
                        must include a worksheet name (eg. Sheet1!A1:B20)
-pedantic             prints error message if types don't match; without this option
                        numbers and datetimes are compared
-s                   silent mode with only errors being printed
-emu                  emulates a standard application; measures parsing and processing
                        times and memory consumption (only Microsoft Windows) without
                        looking for differences
-cache               parses cached formula results instead of formulas; must be used
                        with -emu and is ignored otherwise

C:\Program Files\Pagos Spreadsheet Component for .NET 4>
```

Ssdiff can help finding compatibility problems between Excel and PSC. There can be problems when loading an Excel file and when processing formulas. When you want to test an Excel file the first time just start ssdiff and pass the pathname of the Excel file as the only argument:

```
ssdiff.exe Testfile.xls
```

You will see ssdiff trying to open the file, iterating over worksheets and processing formulas. Ssdiff will show you any differences found between Excel and PSC.

If ssdiff prints an error message you have to find out what causes the error. If ssdiff aborts while trying to open the file, use this command next:

```
ssdiff.exe Testfile.xls -emu -cache
```

Ssdiff will now skip parsing formulas. The option `-cache` makes ssdiff to read cached values only (the results of formulas stored by Excel). As `-cache` can only be used together with `-emu` we have to add this option, too.

If opening the file fails again it is highly possible that the problem is caused by a defined name.

If the file can be opened the error is probably caused by a formula somewhere in the workbook. Unfortunately it is not possible now with ssdiff to find out which formula in which cell actually causes the problem. It is recommended to drop one worksheet after the other until ssdiff can open the Excel file. Then it is known in which worksheet the problematic formula is located.

If the Excel file can be opened but processing formulas causes a problem you can use the `-r` option to process only certain cells. The following command processes only the formulas in the range A1:Z100 of the worksheet Sheet1:

```
ssdiff.exe Testfile.xls -r Sheet1!A1:Z100
```

If the formula in cell M10 causes problems you can only process this formula with this command:

```
ssdiff.exe Testfile.xls -r Sheet1!M10
```

If you know that a formula in a certain cell returns a different result in PSC than in Excel check if it is a nested formula. Try to make it less complicated. Resolve references yourself by passing values directly to formulas. Remove a nested formula and pass the value which you expect to be returned by the nested formula directly. After every change start ssdiff again to process the respective formula. This will help you to narrow down the problem to the exact formula and arguments which cause it.

7. Deployment

As PSC is a component which is integrated and used by other applications these other applications must be able to access the files PSC is made of. We are talking about link libraries – DLLs in Windows – which play together to provide all the functions PSC is known of. Of course the link libraries of PSC must have access to the XML configuration file when PSC is instantiated.

By default all DLLs of PSC are installed to the application directory. When you build an application which makes use of PSC you must make sure that the DLLs of PSC 4 can be found by your application when you start it.

The setup file of the .NET edition can add the application directory to the system variable PATH. There is a checkbox in the setup wizard which you can activate – it is unchecked by default. The reason it is unchecked is that you must not add an application directory to the system variable PATH if you install both the .NET and Java edition. Otherwise both your .NET and Java applications will try to use either PSC for .NET or PSC for Java – one of the two application directories is obviously the first one in PATH. However either your .NET or Java applications will fail then as for one of the two the license won't validate. If you do not

plan to install both the .NET and Java interface you can activate the checkbox and let the setup wizard update the system variable PATH.

For detailed information how Windows resolves DLL dependencies see http://msdn.microsoft.com/library/en-us/dllproc/base/dynamic-link_library_search_order.asp?frame=true.

.NET applications usually install all DLLs they depend on in their application directories. If your .NET application is a Windows program simply copy all DLLs of PSC to the application directory.

Server applications which are built with ASP.NET require a different approach. They usually want DLLs to be installed in the subdirectory bin of the project directory. As most DLLs of PSC are however native Windows DLLs not based on the .NET framework it won't work to copy all of them to the subdirectory bin.

When you deploy an ASP.NET application copy Pagos.Spreadsheet.NET.dll to the subdirectory bin. All other DLLs of PSC must be in a directory which is listed in the system variable PATH.

When you deploy an ASP.NET application you can either update the system variable PATH yourself and add the application directory of PSC or copy all DLLs of PSC (except Pagos.Spreadsheet.NET.dll) to a directory which is already in PATH. Typically the Windows directory is in PATH so you could copy the DLLs there. However be very careful if you have another edition of PSC installed.

8. Uninstallation

PSC 4 can be uninstalled with Add or Remove Programs in the Control Panel. When PSC is uninstalled all files in the application directory are removed except the XML configuration file.

The XML configuration file is not removed as it does not only store configuration details but also the license. To prevent end users from losing their license accidentally they must remove the XML configuration file and the application directory of PSC manually.

As the XML configuration file is not removed when uninstalling PSC it is reused when the same edition of PSC which was removed is installed again to the same directory. The XML configuration file is not overwritten. If it contains a license PSC can be reused immediately again after installing.

If you uninstall the Java or .NET edition of PSC and told the setup wizard to update the PATH variable when installing PSC you must reset the PATH variable yourself. Double-click on System in the Control Panel. You will find the system variables in the registry card Advanced.

9. Known Restrictions

The following list contains known restrictions:

- External references to other workbooks are not supported.

- 3D ranges are not supported.
- Local names are not supported.
- Encrypted workbooks are not supported.
- SQLOPEN and SQLGET are not processed in Excel.