



# USER MANUAL

Smart Packager

USER MANUAL

## Contents

Contents .....	3
Introduction .....	4
Configuration.....	5
CVS Roots .....	5
Add repository.....	5
Remove repository .....	7
Scripts .....	7
Scripts SQL – Order and Divide .....	7
Pre order .....	8
Pre division .....	9
Division by tag .....	9
SQL Scripts – Installation Scripts .....	9
SQL Scripts – SQL Files.....	10
Features.....	11
Work Directory.....	11
Directory to the CVS checkout .....	11
Default Package Name .....	11
Default Script Name .....	11
Conflict resolution method .....	11
Information Register .....	12
Send anonymous data usage .....	12
Packaging.....	13
Get files .....	14
Add modules .....	14
Add tag .....	15
Group Files .....	17
Scripts operations.....	17
Placing scripts.....	18
Files information .....	18
File operations.....	19
Manually order files .....	19
Drag and Drop .....	19
Package name .....	19
Package created .....	20
Tag files .....	20
Open package folder .....	20
Finish wizard.....	21
Licensing.....	22
Legal Information .....	24

Introduction

This manual describes how to use Smart Packager at user level. It is target to technical users, but it is necessary to configure and to integrate with project’s versioning tool.

Please see our website for more information on release notes and available product sheets.

This chapter contains the following sections:

Configuration..... 5

Packaging..... 13

Licensing..... 22

Legal Information ..... 24

## Configuration

Configuration dialog allows you to define several application parameters, including repositories where files to be packed are stored, pre processing (sorting and division) of SQL files and other overall parameters.

### CVS Roots

The CVS roots tab stores CVS repository configurations from where files are obtained. This tab displays a set of attributes that define CVS Roots characteristics and the list of all configured roots.

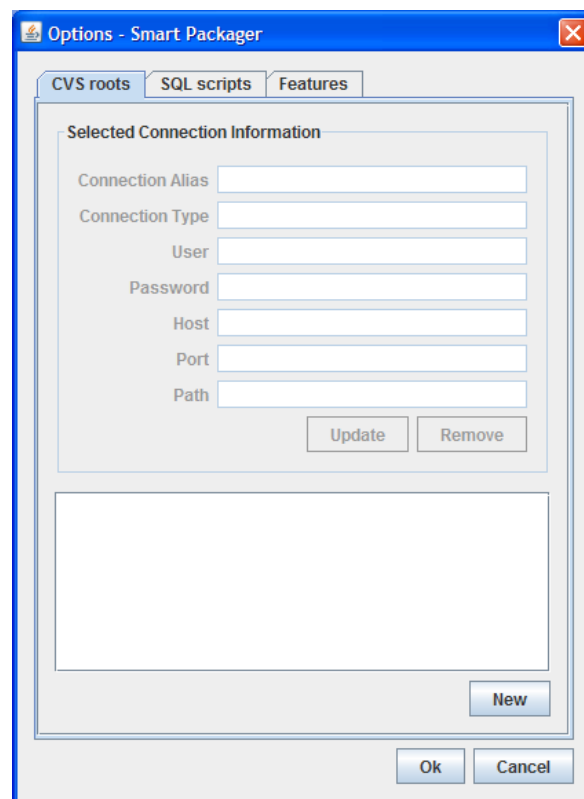



Fig. 1 – CVS Root

### Add repository

Clicking  button will activate the CVS root form where you can set all necessary CVS root attributes.

New Connection Information

Connection Alias

Pandora

Connection Type

pserver

User

jsmith

Password

.....

Host

pandora

Port

Path

/var/cvs/dev

Add

Cancel

Fig. 2 – CVS Roots – Add repository

By adding the form information, a new repository will be available in the repositories list below the form. Several repositories can be added as necessary.

pserver:jsmith@pandora:/var/cvs/dev

Fig. 3 – CVS Roots – Repository list

Selecting an element in the repository list will fill the form with its attributes. After this any of the repository attributes can be modified and changes can be confirmed by clicking the 

Update

 button.

Selected Connection Information

Connection Alias

Pandora

Connection Type

pserver

User

jsmith

Password

.....

Host

pandora

Port

Path

/var/cvs/dev


Update

Remove

pserver:jsmith@pandora:/var/cvs/dev

Fig. 4 – CVS Roots – Update repository

## Remove repository

Repository removal is done by clicking the  button that becomes available after selecting an element in the repository list. When you remove a repository, a confirmation window is launched, as show below.

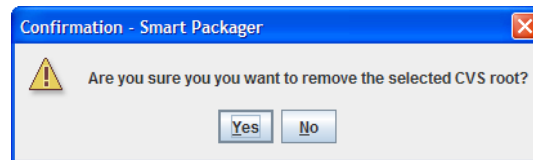


Fig. 5 – CVS Roots – Remove repository confirmation

## Scripts

### Scripts SQL – Order and Divide

This tab has a tree of folders that let you set a pre order and division of SQL files and definition of the base SQL scripts by repository tag.

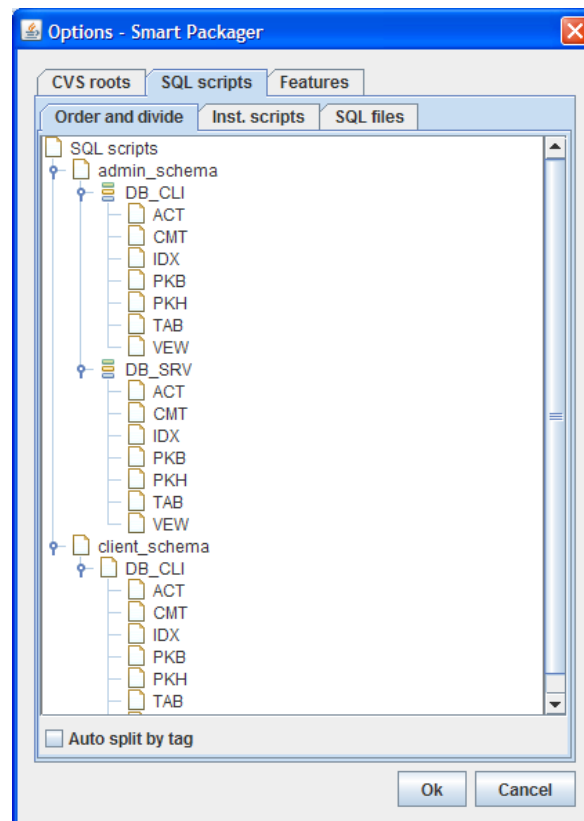


Fig. 6 – Scripts SQL

The tree must map the folder where the files reside in the repository. The mapping is made from the leaves to the root. You will not need to replicate the entire folder structure from the repository, but only the structure (from the leaves to the root) that can define the ordering and splitting of files.

In this tree there is a popup menu available and it is activated with the right mouse button.

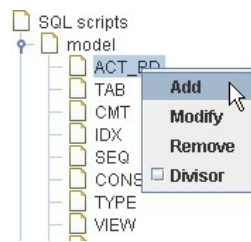


Fig. 7 – Scripts SQL – Popup menu

The popup menu provides the following operations on the nodes of the tree:

- Add** ..... Adds a new node within the selected node;
- Modify** ..... Modify the name of the selected node;
- Remove** ..... Remove the selected node;
- Divisor** ..... Change the node type to divisor (see **Erro! A origem da referência não foi encontrada.**).

### Pre order

Database files will be installed based on the tree's hierarchical sequence. Each node's relative position matters to the final installation order.

It is possible to reorder the tree by simply dragging its nodes. If, while dragging, the node is released in an area between two other nodes, it will take that position.

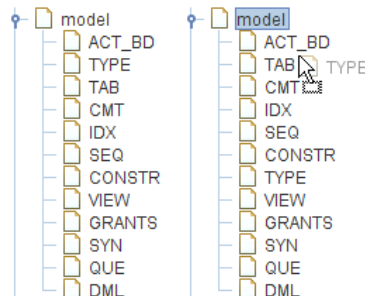


Fig. 8 – Scripts SQL – Move items

If dropped on top of another, item is placed within that item, defining a hierarchy.

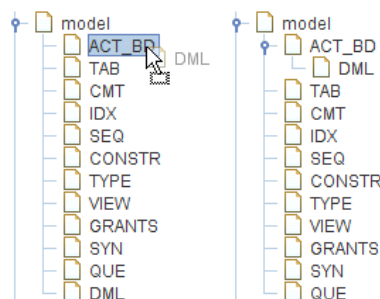


Fig. 9 – Scripts SQL – Hierarchy items

While pressing CTRL key, you can execute the two prior movements to copy the selected item.



### Pre division

The pre division process allows the creation of installation scripts divided by folders that can, for example, identify the scripts type. Besides the division by folders, the scripts order is still correctly maintained, which originates a bigger fragmentation of the installed scripts.

Divisor items ensure not only that the software within these items runs on single scripts, but also that the software within the previous items is separate from the items that follow.

Divisor nodes are easily identified by their icon.



Fig. 10 – Scripts SQL – Non divisor and divisor

### Division by tag

Besides the pre division according to the structure of the repository folders is still possible to make an additional division by tags. This behavior is enabled or disabled at the bottom of the SQL scripts configuration tab.

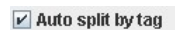


Fig. 11 – Scripts SQL – Split by tags

## SQL Scripts – Installation Scripts

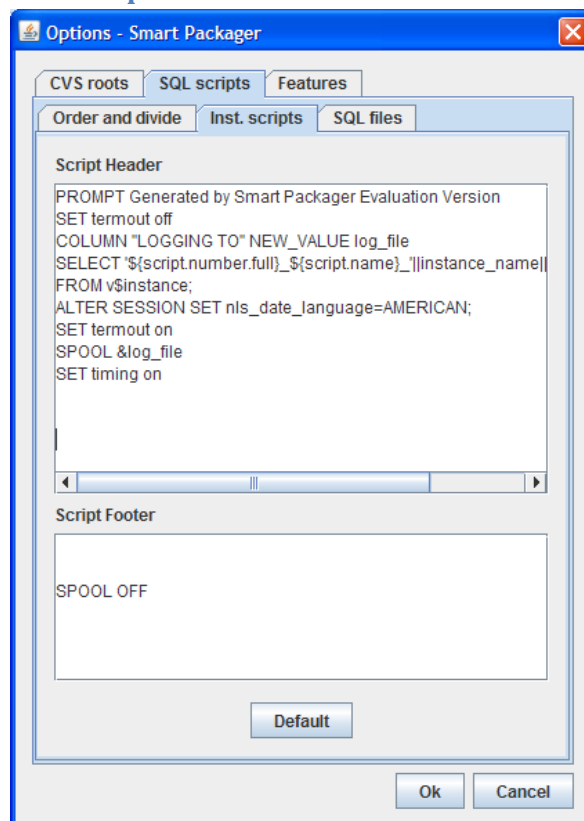


Fig. 12 – Scripts SQL – Inst. Scripts

This tab is possible to edit the header and footer of installation scripts. It's possible to define system variables described below:

- `${script.name}`      this is the script name;
- `${script.number}`      this is the script order number generated;
- `${script.number.full}`      this is the script order complete number 'XXX';

## SQL Scripts – SQL Files

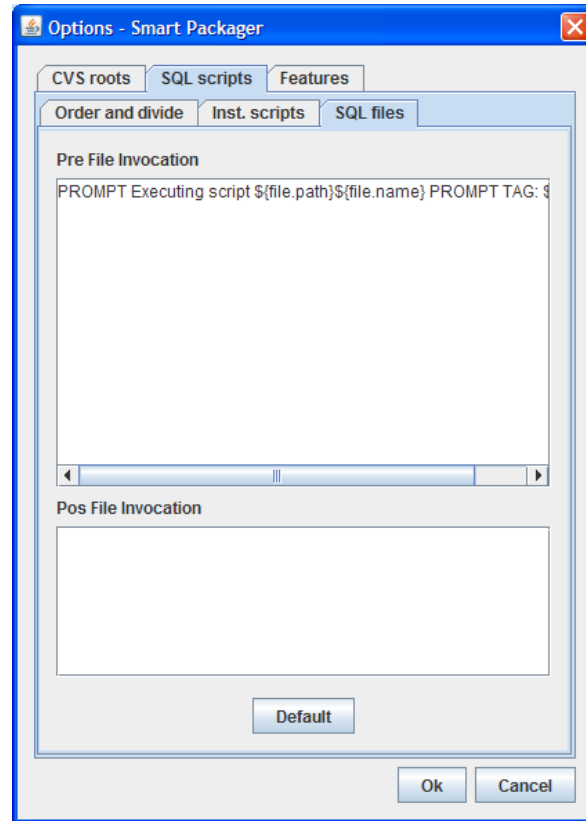


Fig. 13 – Scripts SQL – SQL Files

This tab is possible to edit the header and footer of the files in the installation scripts. It's possible to define system variables described below:

- `${ file.name}`      this is the name of the SQL file;
- `${ file.tag}`      this is the name of the tag from the SQL file;
- `${ file.path}`      this is the complete path of the SQL file;

## Features

On the features tab there are some global parameters that may change the application behavior. These parameters are described below.

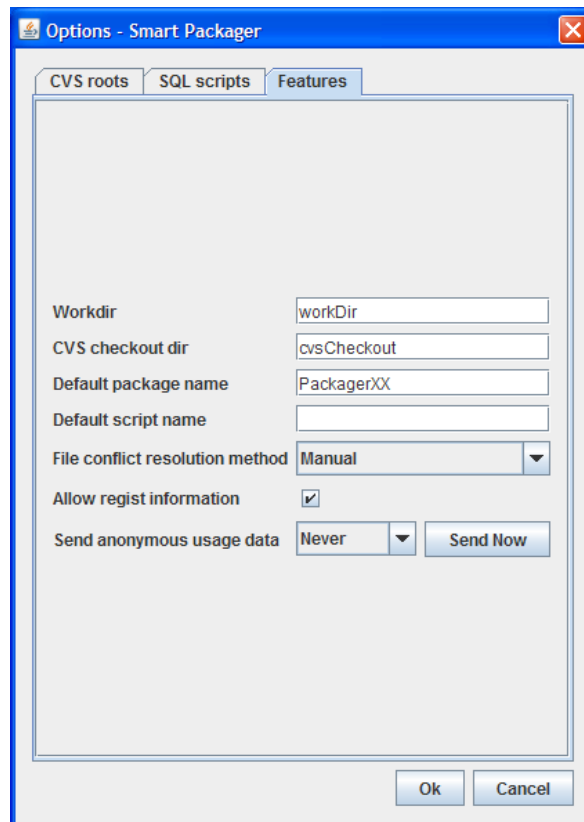


Fig. 14 – Features

### Work Directory

This field defines the application's work directory. This directory can be absolute or relative to the application's execution directory.

### Directory to the CVS checkout

This field sets the directory within the working directory, which is used to store the files obtained from the repository.

### Default Package Name

When it isn't set any name for the package being created, the name that is defined in this attribute will be used.

### Default Script Name

When a script is added the name that is defined in this attribute will be used. It's allowed to change its name after.

### Conflict resolution method

Identifies the method used by conflict resolution algorithm. There are three options which behaviors are described below:

**Manual** ..... Is not done any processing for conflict resolution, and the user has to perform this operation manually;

**By date** ..... The conflict resolution is based on the file's repository date and kept the file with most recent date;

**By revision**.... The conflict resolution is based on the file's repository revision and kept the file with the newer revision.

When the method of conflict resolution is not manual and any conflict is resolved, informative logs appear in the logs area.

### Information Register

When this field is activated the application start to store logs. These logs are then saved to a file when the application closes. If the field is deactivated the file, if exist, is deleted from the computer.

### Send anonymous data usage

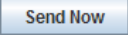
This field is only available when the preview option is activated. This option allows the application to send the data stored in the file to the server, based on the fooling time periods:

**Never**

**Daily**

**Weekly**

**Monthly**

The user can also send this data to the server manually by clicking on . The application will then try to send the file to the server and delete the file if successfully sent.

This info allow the developers of Smart Packager to find the reported bugs more easily to be found, also it will allow to optimize the application based on the most frequently used operations.

## Packaging

The packaging is done in a three steps wizard:

**Checkout Files** ..... Obtains files from one or more repositories, through the specification of the required repository modules and tags;

**Group Files** ..... After obtained from repository, files are pre ordered and divided into scripts. The result of this pre process can always be manually changed. By entering this stage, automatic conflict resolution can also be made;

**Package Created** ..... At this point the package has been built on the existing settings and changes made manually. It is also possible to tag packed files.

The application window is divided into 5 areas described below:

- 1 ..... Wizard's indicator panel;
- 2 ..... Work area;
- 3 ..... Wizard's buttons;
- 4 ..... Task's progress bar;
- 5 ..... Logging area. It is an expandable area that, by default, is collapsed.

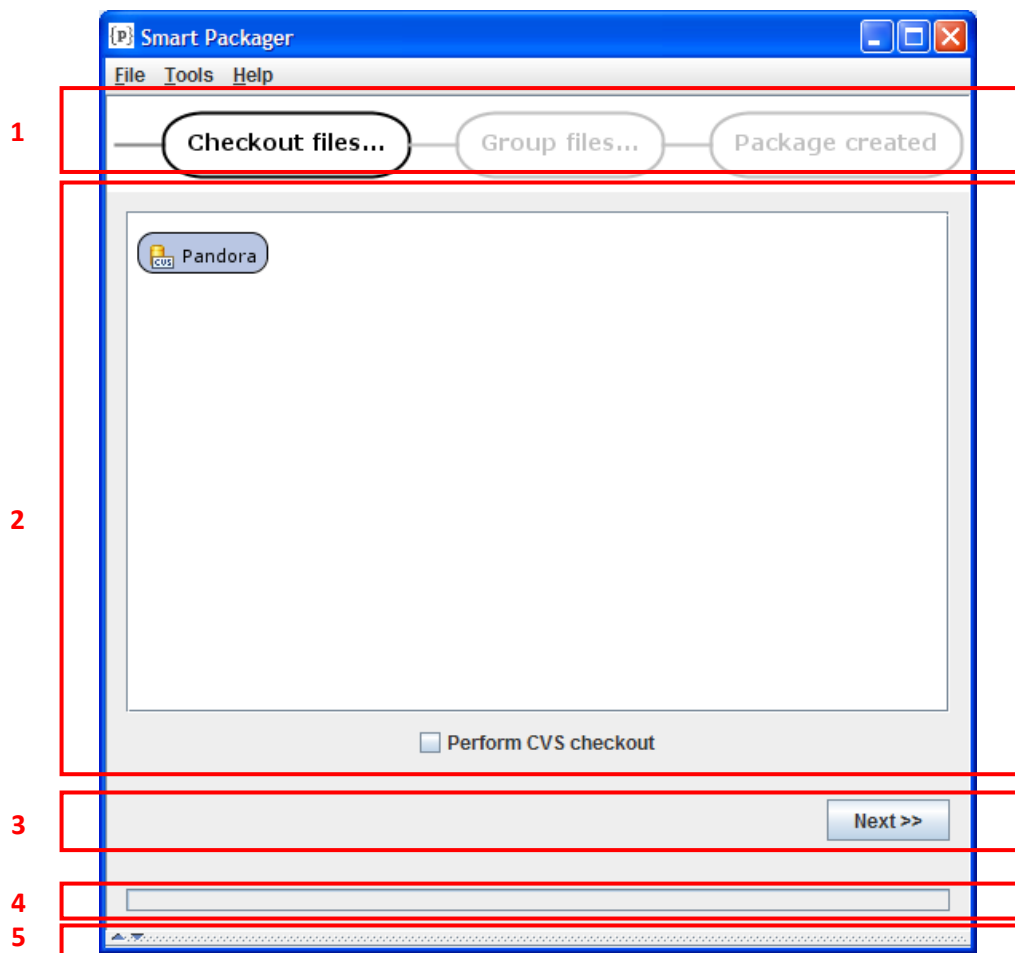


Fig. 15 – Smart Packager – Global Components

## Get files

The checkout files panel is open when application starts and you can specify whether to obtain files from repository or use the previously checked out files. To be able to obtain the files you will need to define their modules and tags and select the checkbox at the bottom of the work area.

It may be useful to leave the CVS checkout option unchecked, if this task was already performed before and you already have all the files you want to pack.

## Add modules

If you want to retrieve files from the repository, you need to define their modules in the work area. To do this, you must click the right mouse button on top of the repository to open the corresponding popup menu. This menu only allows you to add modules.

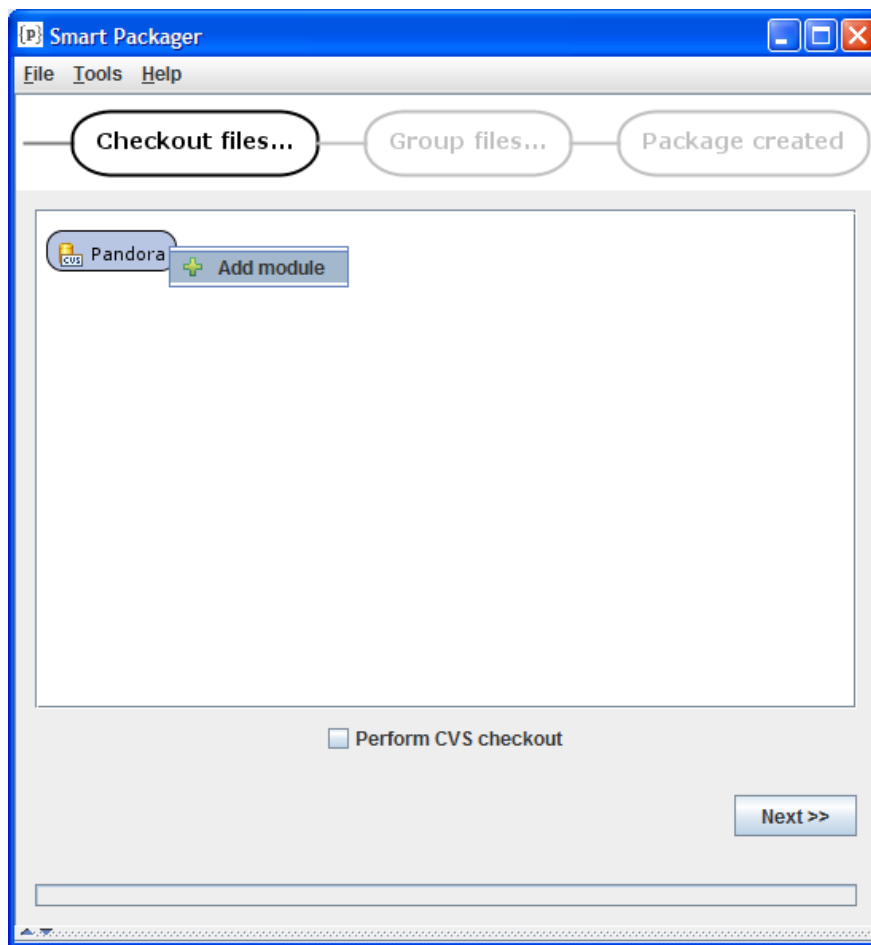


Fig. 16 – Checkout Files – Add modules

Selecting the existing option will launch a window where is requested the name of the module to be inserted.

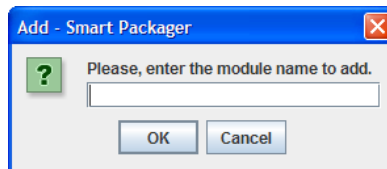


Fig. 17 – Checkout Files – Add module

### Add tag

After adding the module, it will be necessary to specify the tags to retrieve files from repository. This is possible by launching the popup menu for the desired module. You can also use this menu to change the name of the module or remove it.

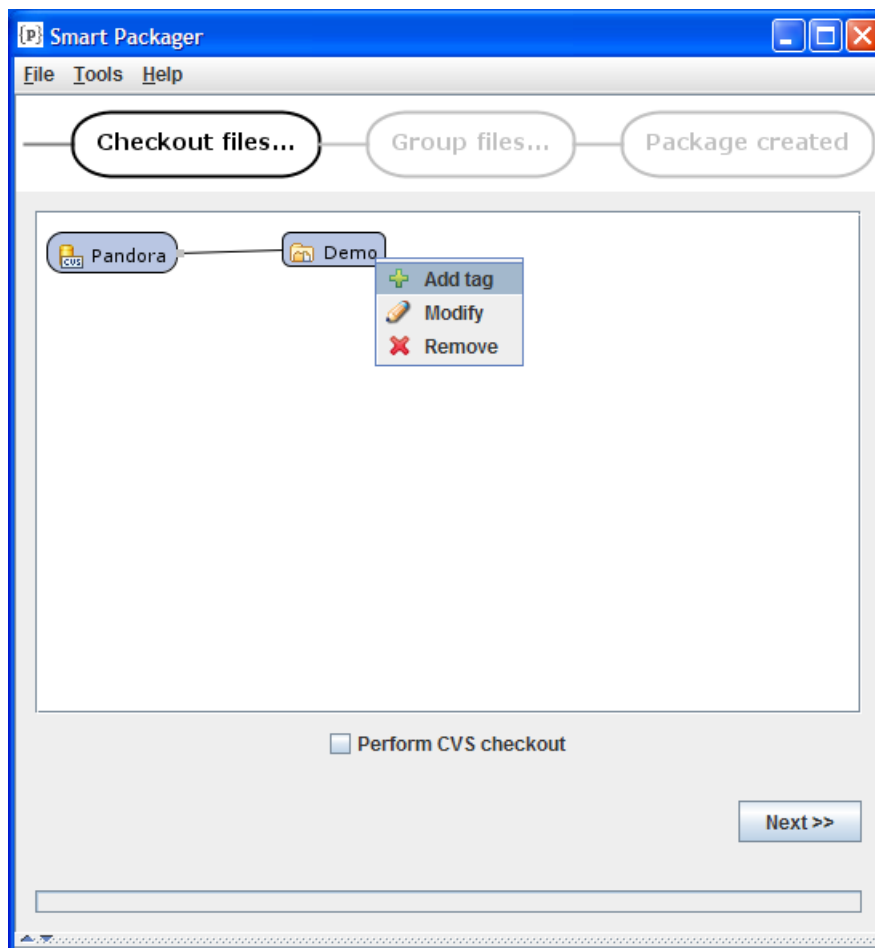



Fig. 18 – Checkout Files – Add tag

The option of adding tags launches a window for that purpose. In this window you can manually specify the tag that you want to add or retrieve a list of tags available for the specified module, by clicking .

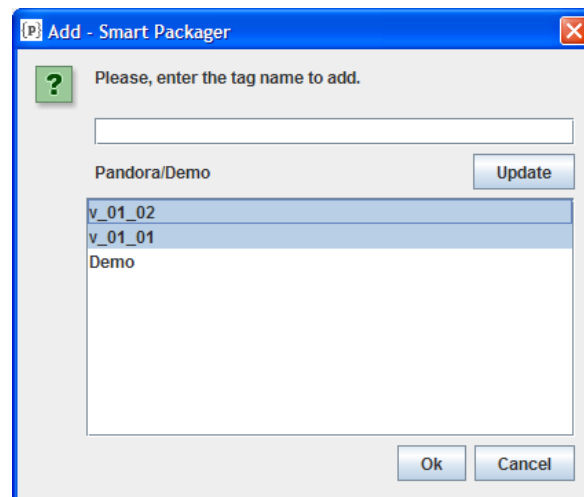


Fig. 19 – Checkout Files – Add tag Window

After the list is updated, one or several tags can be selected and added to the module.

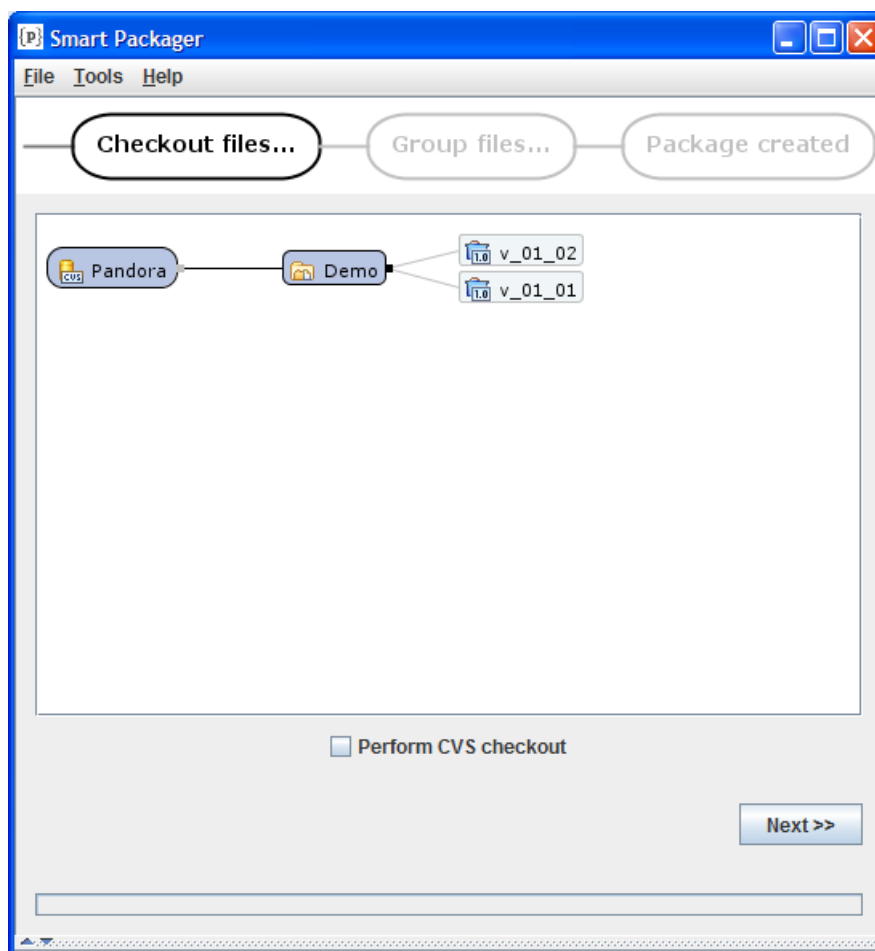


Fig. 20 – Checkout Files – Complete definition of modules and tags

The tags also have a popup menu that allows to change the name or to remove the selected tag.



With all modules and tags specified, click the **Next >>** button to obtain files from repository, pre-process them (pre order and divide) and automatically resolve file conflicts.

## Group Files

The file grouping panel has an area where the package preview is graphically displayed. On the right side of the main area are the files obtained from the repository and on the left are the scripts that will install them. The files were previously ordered and divided into scripts that will install them.

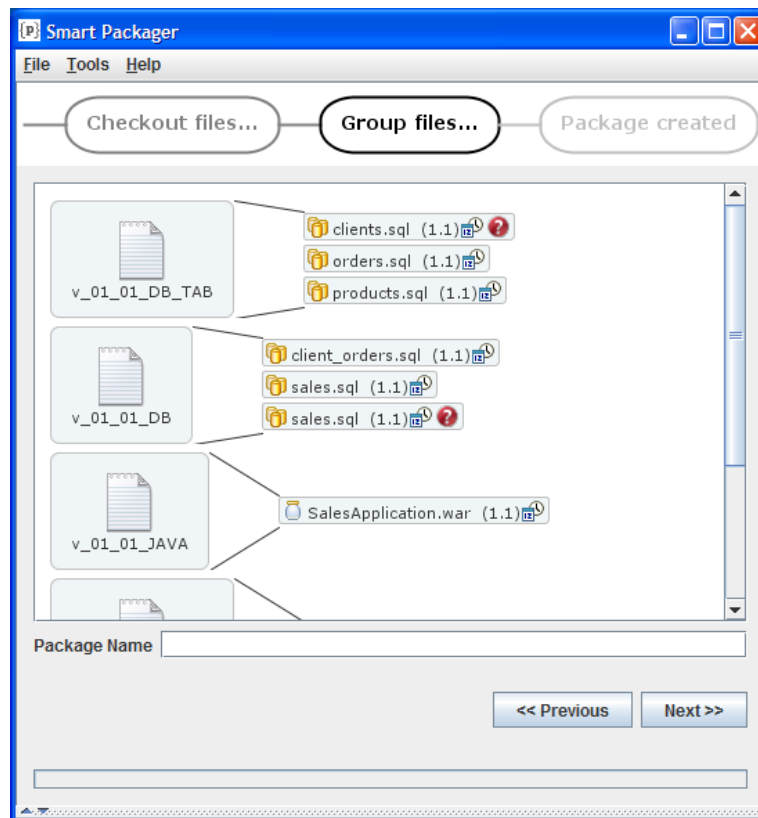


Fig. 21 – Group Files

## Scripts operations

In this panel there is available a set of operations which can change the suggested order and division.

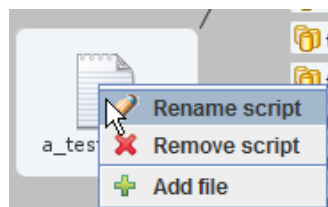


Fig. 22 – Group Files – scripts operations

The available operations for scripts are described below:

**Rename script** ..... Allows renaming the script generated for the package;

**Remove script** ..... Allows removing a script and all associated files;

**Add file** ..... Allows adding an external file to the script. Selecting this option will launch a window to select the desired file.

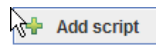


Fig. 23 – Group Files – add script

Clicking on an empty spot allows the following:

**Add script** ..... Allows adding a new empty script to the end of the list;

### Placing scripts

In addition to these operations is still possible to manually change the scripts order by simply dragging it to the desired position.

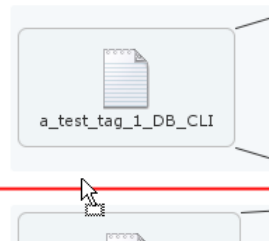


Fig. 24 – Group Files – Move scripts

### Files information

The files have more information than what is initially visible. The file component has the format specified below and some of these sub-components may or may not be present.

- 1.....File type icon. Clicking this icon, file name (2) is expanded or contracted in order to distinguish files with same names but in different locations;
- 2.....File name or full path of the file;
- 3.....File revision from the repository. This part is not present when the file is added manually;
- 4.....Calendar icon. Clicking this icon, date and time of the file revision is expanded or contracted. This part is not present when the file is added manually;
- 5.....Conflict warning icon. The presence of this icon indicates that the file has conflicts, for example, the same file was obtained with different tags. Clicking this icon will link it with all files that are in direct conflict with it. This part is only present when there are unresolved conflicts.

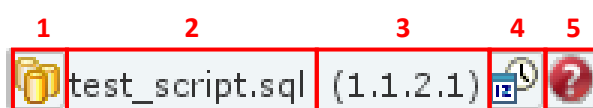


Fig. 25 – Group Files – File format

By double-clicking over a file you can open it with the default system editor for this kind of files.

### File operations

The remove file operation is only available on files (not scripts).



Fig. 26 – Group Files – File operations

### Manually order files

As with the scripts, you can change the position of files. The files can be moved one or several at a time. The selection of multiple files is done using the keyboard and mouse, where CTRL and SHIFT work as follows:

**CTRL** ..... Add or remove the file by clicking on its component;

**SHIFT** ..... Extend the selection of the file's original selection to the clicked file.

Changing the position of the files is done by drag-and-dropping the selected files to the desired position.

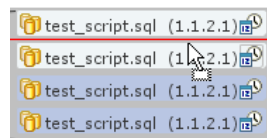


Fig. 27 – Group Files – Move files

### Drag and Drop

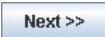
The application allows dropping files from windows to the installation scripts, the user can simply select a file or multiple files from windows, drag them to the application, and drop on the desired location.

### Package name

There is a text box at the bottom of the work area that can be filled with the package name.



Fig. 28 – Group Files – Package name

When this field is left empty the default value for this field will be used (see Default Package Name) although you will be prompted for confirmation after you click the  button.

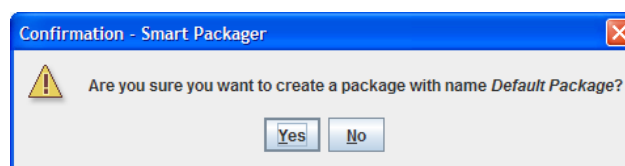


Fig. 29 – Group Files – Default package confirmation

## Package created

At this point the package has been created with the specification made, either in the settings, either manually from the file grouping panel.

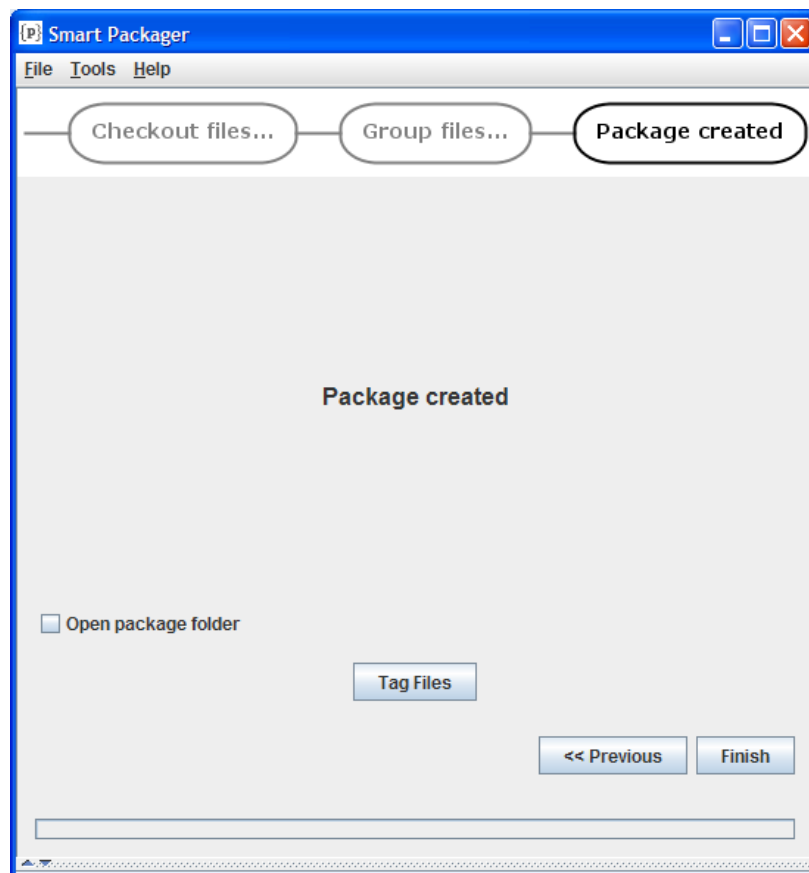


Fig. 30 – Package created

## Tag files

To tag the packed files with the package name, or any other tag, simply click on . It will launch a window where the desired tag can be entered. By default, the package name is presented.

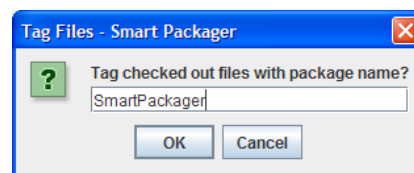


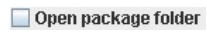
Fig. 31 – Package created – Tag files

All files in the package are marked with the chosen tag after confirmation. Files obtained from repository but not packed are excluded from this tagging, as well as all the external files added manually.

## Open package folder


The checkbox in this screen allows, at the conclusion of the wizard, to open a file explorer window showing the folder where the package was created. This feature is interesting in case

you want to make any subsequent changes to the package, copy the files generated to any other location or compress the file into an archive.



**Fig. 32 – Package created – Open package folder**

### **Finish wizard**

Completing the wizard is done by clicking the  in this last panel.

## Licensing

When the application is executed and there isn't a valid certificate (the current certificate has expired, is corrupt or has never been obtained) a wizard will be launched to get a new certificate for the application.

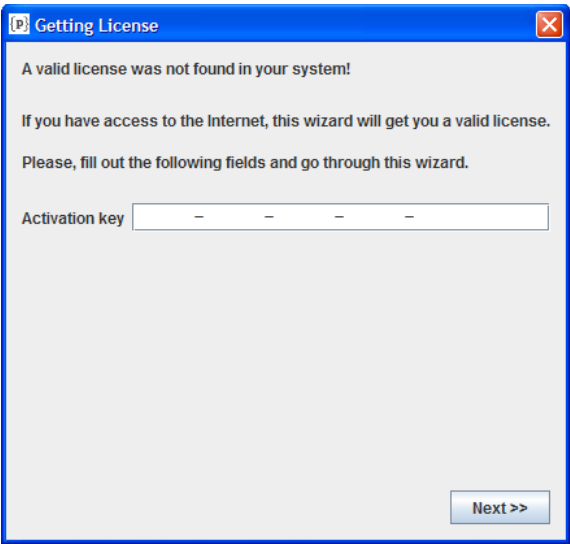



Fig. 33 – Certificate application

In this window, the license key and user name should be inserted in order to carry on with the certification process. Clicking the  button will submit the information to Shortcut's server, where it will be verified and the corresponding certificate will be issued. This certificate proves the authenticity of the client.

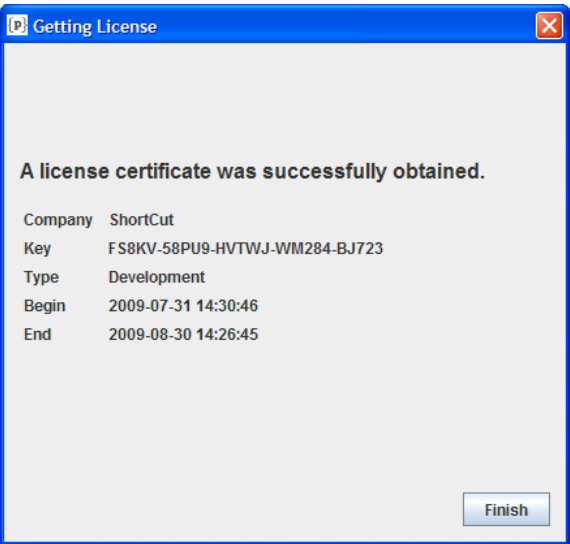


Fig. 34 – Certificate application – Certificate successfully obtained

The certificate will be stored in a file along with the other application files and should not be removed.

If the server cannot be contacted, a validation token will be generated which can be sent to the indicated email in order to manually complete the same process.

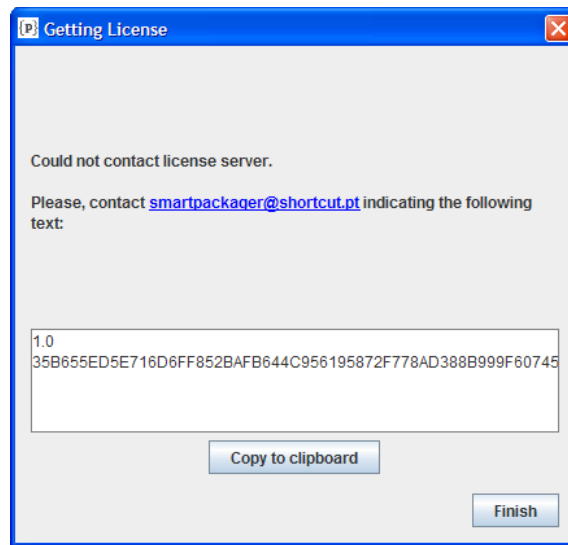


Fig. 35 – Certificate application – impossible to contact the server

The support team will manually validate the token and send back an email with the certificate for the requested application. The certificate should then be placed next to the application files.

## Legal Information

2009 Shortcut, Lda. All rights reserved.

Smart Packager's User Manual.

This manual, as well as the software described in it is furnished under license and may be used or copied only in accordance with the terms of that license. The contents of this manual is provided for informational purposes only, is subject to change without notice and should not be construed as a commitment by Shortcut Limited. Shortcut assumes no responsibility for any errors or inaccuracies that may occur in this documentation. Except as permitted by such license, no part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, recording or otherwise, without the prior written permission of Adobe . Remember that artwork or images will be included in your existing project may be protected by copyright. The unauthorized incorporation of such material to their work may be a violation of the copyright owner. Make sure you get the required permission from the owners of copyright. Any reference to company names in sample templates intended only for demonstration purposes and is not intended to refer to any actual organization.

Shortcut, Lda. – Avenida Serpa Pinto, 508 – 4º piso 4450-277 Matosinhos