

Configuration / API in PSNext 3.0

02/15/09 – Revision 1.0

This document, as well as the software described in it, is furnished under license and may only be used or copied in accordance with the terms of such license. The information in this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Sciforma. Sciforma assumes no responsibility or liability for any errors or inaccuracies that may appear in this document. No part of it may be reproduced or transmitted, in any form or by any means without the prior written permission of Sciforma. Copyright 2009.



Sciforma Europe: 10 rue Vercingétorix 75014 Paris – France
Tel : +33 (0) 1 56 54 83 00 – www.sciforma.fr – info@sciforma.fr

Content

A -Introduction	3
B -Config Utility	4
1 -Selection of available languages.....	5
2 -Force formatting conventions used in PSNext.....	6
3 -Default configuration language.....	7
C -JRE	8
1 -Java 1.5.....	9
2 -Java 1.6.....	9
D -Compatibility	10
1 -JRE tested and supported by Sciforma.....	10
2 -Application servers tested and supported by Sciforma.....	10
3 -Databases tested and supported by Sciforma.....	11
E -API	12
1 -Documentation changes.....	13
2 -Samples changes.....	13
3 -API function changes and additions.....	13
3.1 -The Global category.....	13
3.2 -The com.sciforma.psnnext.api.clientevent package.....	13
3.3 -Accessing constants.....	13
3.4 -New support method for lists.....	13
3.5 -Pooled risk support added.....	14
3.6 -New methods for the Project class.....	14
3.7 -New methods for the Resource class.....	14

A - Introduction

This white paper describes new features and modifications implemented for PSNext 3.0 configuration. Essentially, these modifications concern the Config Utility and the JRE version compatibility.

Detailed technical specifications are listed, as well as API enhancements.

B - Config Utility

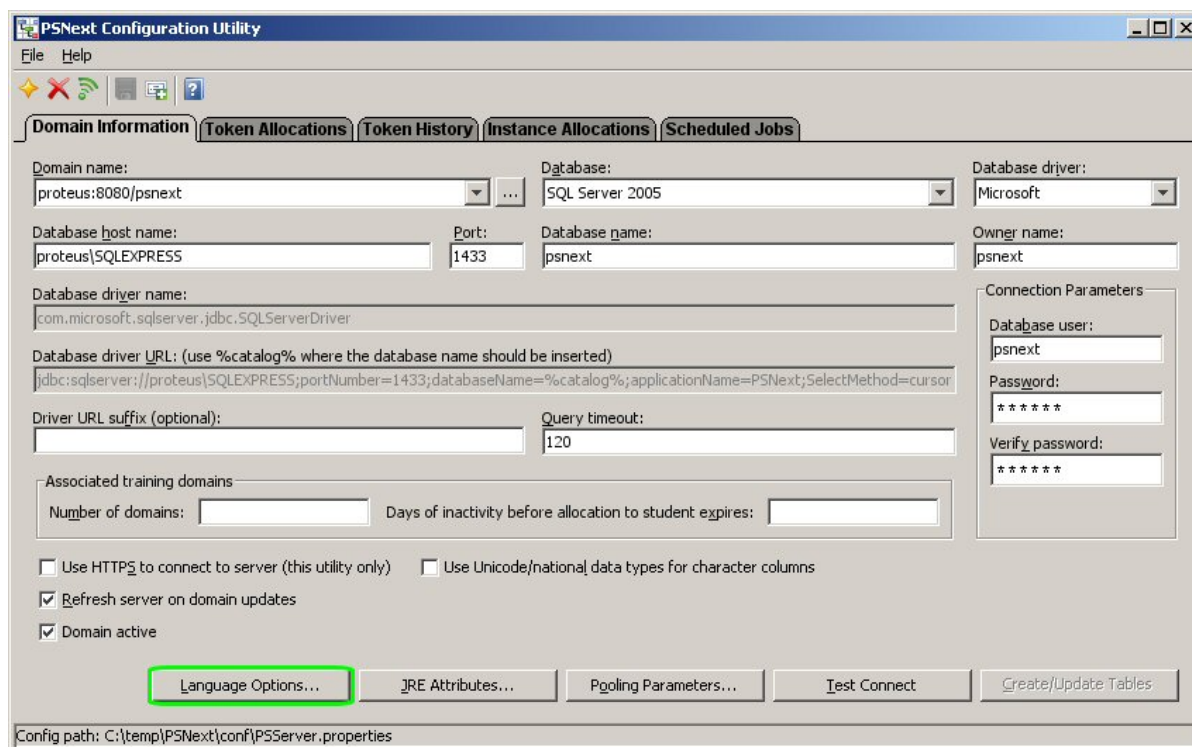
Until now, the PSNext GUI language could be selected by each user in the connection dialog box. This listed all of the languages available in the PSNext version installed on the server. One could not restrict the content of this list to a selection of all these possible languages.

PSNext 3.0 offers a function to specify the list of languages available to users when they log in.

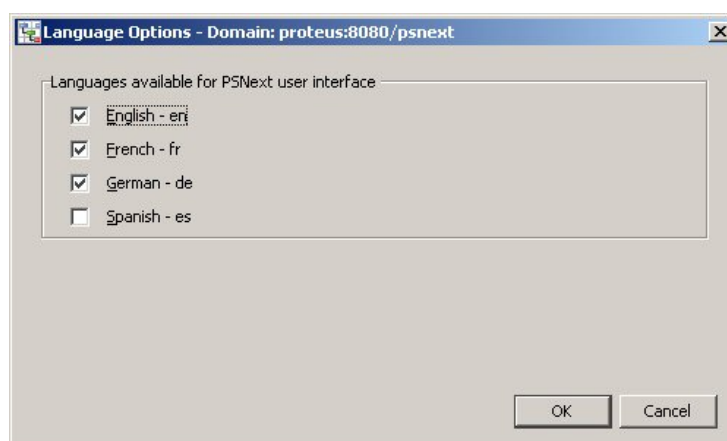
1 - Selection of available languages

To define the list of languages available at connection:

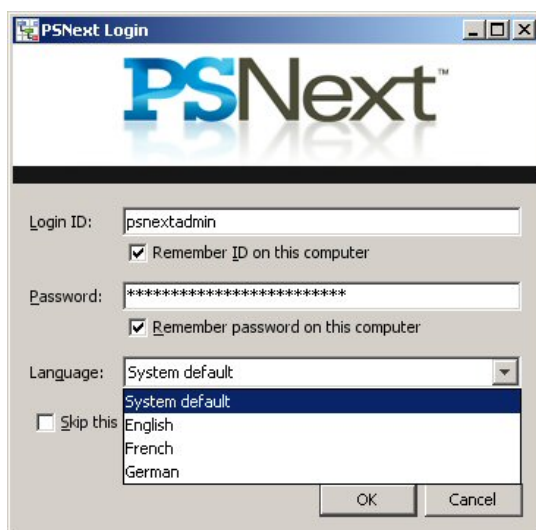
- Launch PSNext Config Utility,
- Click on the "Language options...." button,



- Select the language(s) you want to make available to users – at least one must be selected,



- Validate with the "OK" button,
- Update and active this PSNext configuration by clicking the "Save" button of the configuration utility.
- Start PSNext,



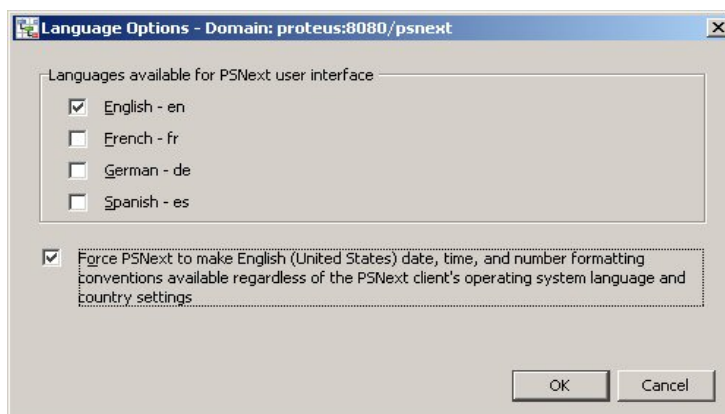
The list of available languages is now limited to those selected in the Config Utility.

2 - Force formatting conventions used in PSNext

PSNext 3.0 offers the possibility to force a unique language for all users and to use corresponding formatting conventions whatever the client workstation regional settings.

To force a unique language for all users and to use corresponding formatting conventions:

- Launch PSNext Config Utility,
- Click on the "Language options...." button,
- Select the unique language you want to make available to users,



If only one language is selected an option shows at the bottom reading "Force PSNext to make [language] date, time and number formatting conventions available regardless of the PSNext client's operating system language and country settings." As it clearly states, it gives the possibility to use specific formatting conventions in PSNext with no respect to the workstation regional settings.

3 - Default configuration language

PSNext setup program installs several configuration files in the PSNext configuration folder. Each of these corresponds to one of PSNext possible languages:

- default_en.cfg (English),
- default_fr.cfg (French),
- default_de.cfg (German),
- default_es.cfg (Spanish).

When the configuration utility is used to create tables at the end of the first PSNext installation, this operation eventually imports one of these language specific configuration files into PSNext.

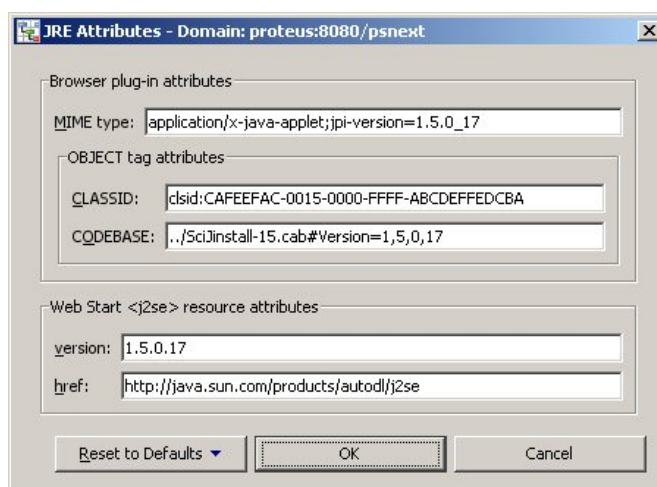
The configuration file is selected according to the operating system language of the workstation where the configuration utility is run. For instance, on a workstation with French regional settings, the default_fr.cfg configuration file will be imported into PSNext when tables are created.

It must be noticed that a configuration file is automatically imported when tables are first created but not if/when they are later updated. After installation, manual configuration files export/import operations may be performed through the Import.../Export.. options of the File menu in the System module in PSNext.

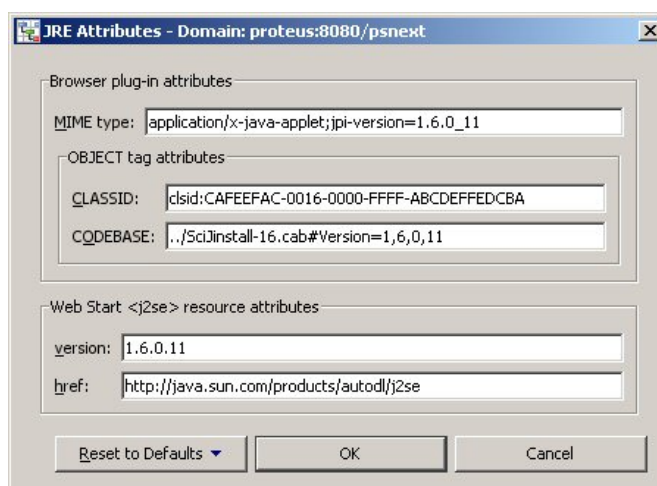
C - JRE

PSNext 3.0 gives the possibility to choose between Java 1.5 or Java 1.6 environments whereas the previous version could be used with Java 1.4 and Java 1.5.

1 - Java 1.5



2 - Java 1.6



The Java version expected on client stations is specified in the configuration utility along with various options for this Java version.

When PSNext is run on a client station and this required Java environment is not installed on this station, PSNext gives the possibility to download and install it (on Windows stations). It's necessary to have administration privileges to install the Java environment.

In case the Java environment should be installed on many workstations, it is recommended to pre-deploy it with the usual deployment tools.

D - Compatibility

1 - JRE tested and supported by Sciforma


- JRE 1.5 (version 1.5.0_11 and later).
- JRE 6 (Version 6u11 and later).
- ⚠ *JRE 1.4.2 is not supported anymore.*

2 - Application servers tested and supported by Sciforma

- Apache Tomcat 5.5 and later.
- BEA™ WebLogic™ Express 8 and later.
- IBM™ WebSphere™ Application Server - Express 6.1 and later, or WebSphere CE.
- Red Hat JBoss Application Server (minimum version 4 or later).

3 - Databases tested and supported by Sciforma

- Microsoft™ SQL Server™ 2000 SP3 (Microsoft™ JDBC driver version 1.2).
- Microsoft™ SQL Server™ 2005 SP2 (Microsoft™ JDBC driver version 1.2).
- Oracle9i™ Database 9.2 (Oracle JDBC driver version 10.2.0.3).
- Oracle10g (Oracle JDBC driver version 10.2.0.3).
- IBM™ DB2™ Universal Database 9 and later (IBM™ JDBC driver).
- PostgreSQL 8.1 and later (PostgreSQL JDBC driver).

 *JNetDirect JDBC drivers are not anymore supported.*

E - API

This paper aims at notifying and explaining the updates regarding PSNext 3.0's API. It will also describe the changes in the sample files and the changes concerning the documentation itself.

We will first start with the samples and documentation concerns before studying the new API functions or updates.

1 - Documentation changes

The ReadMe.txt file, located at the root of the ApiSamples folder, has been extended with notes 7 and 8, explaining the changes done in the .bat files which build the sources from the samples (see Samples changes for more information).

2 - Samples changes

Some modifications have been brought to the script building the examples. It now builds the sample presenting the server-side event listener system, as well as a sample aimed at demonstrating the UI Commands. Keep in mind, though, that building these two new examples only creates the archives, and does not prevent from having to configure and install the « custom.properties » file (located at the path com/sciforma/psnext/api) in the case of the server events, nor from having to create the UI Command in PSNext for the UI Commands sample.

One more source file has been added, UICommandsSample.java, containing a sample code for designing a UI Command.

3 - API function changes and additions

3.1 - THE GLOBAL CATEGORY

The API in PSNext 3.0 has been enhanced with a completely new access to the Global type objects, corresponding to the Global datatype in PSNext. You can create such an object by using the constructor, and accessing Global field just the way you would access any other object's fields.

3.2 - THE COM.SCIFORMA.PSNEXT.API.CLIENTEVENT PACKAGE

This new package contains the listener and the event class related to the new UICommands system.

3.3 - ACCESSING CONSTANTS

Though the constants are still accessible as they were before, they are now inherited from « container » objects, such as the ConfigCategories class or the TimeIntervals class. It can be an alternate way to access these in a generic way.

3.4 - NEW SUPPORT METHOD FOR LISTS

Text lists fields can now be accessed through the method getListField(fieldname) and set through the method setListField(fieldname, List).

3.5 - POOLED RISK SUPPORT ADDED

Two classes have been added to enable the support of pooled risks through the API. The first class, `PooledRiskLevelResourceOptions`, aims at setting the parameters of the leveling, such as the starting date, the finish date, whether the tasks are splittable or not, methods to get or define if availabilities should be scaled or not and how... The second class, `PooledRiskSimulationOptions`, enable to get or define the confidence rate of the simulation, the duration formula, if the resources must be leveled or not... The `PooledRiskLevelResourceOptions` is used mostly to encapsulate the datas representing the resource options. It's not used on its own.

The `PooledRiskSimulationOptions` is not meant to be used by itself either ; when you defined your options, you have to call the Project level method « `runPooledRiskSimulation(options)` ».

For more information on the risk simulations, please refer to the functional White Papers.

3.6 - NEW METHODS FOR THE PROJECT CLASS

The Project class has been added a few new methods :

- `Project copy()` : this method returns a Project object, identical to the calling object.
- `boolean hasChanged()` : indicates if there's been any change applied to the project since it last got saved or since it was opened.
- `void runPooledRiskSimulation(PooledRiskSimulationOptions options)` : calls a risk simulation scenario.

3.7 - NEW METHODS FOR THE RESOURCE CLASS

The Resource class got a new save method : `void save(boolean unlock, boolean publish)`. It enables to define whether or not you want to publish the saved resource, which was not possible in the previous version of the API.