



# PLUG-IN CERTIFICATION PROCEDURE

All Rights Reserved.

No part of this publication may be reproduced, transmitted, transcribed, or stored in a retrieval system in any form by any means without the written permission of SWsoft..

The information contained herein is privileged, confidential and intended for the use of the intended recipient. The Receiving Party agrees to receive and hold Confidential Information, whether transmitted orally, in writing or in any other form, and whether prepared by a party or its Representatives, in confidence, and to use the Confidential Information solely for the purpose of evaluating the proposed business relationship with the Disclosing Party. The Receiving Party further agrees that all Confidential Information will be kept confidential by itself and its Representatives and shall not be disclosed to any third party without the prior written consent of the Disclosing Party; provided, however, that any of such information may be disclosed to a Receiving Party's Representatives who need to know such information for the purpose of evaluating any such possible transaction between the parties (it being understood that such Representatives shall be informed of the confidential nature of such information and shall be directed to treat such information confidentially). The Receiving Party agrees to be responsible for any breach of this Agreement by any third party to whom the Receiving Party has provided Confidential Information, or any portion thereof.

While every precaution has been taken in the preparation of this document, SWsoft, Inc. assumes no responsibility for errors, omissions, or for damages resulting from the use of the information herein.

HSPcomplete is a registered trademark of SWsoft, Inc. All other trademarks are property of their respective owners

---

## 1 INTRODUCTION

---

### 1.1 OVERVIEW

This document describes the HSPcomplete plug-in certification procedure. Potential applicants should make sure that third-party plug-ins are fully compatible with the SWsoft HSPcomplete platform and to fully use the HSPcomplete SDK found on the SWsoft Developer Network (<http://swdn.swsoft.com>)

### 1.2 TERMS

Terms of the certification and linking agreement can be found at <http://www.swsoft.com/en/products/hspcomplete/plugin/agreement>

### 1.3 HOW TO APPLY FOR CERTIFICATION

In order to apply for a plug-in certification please submit application form at <http://www.swsoft.com/en/products/hspcomplete/plugin/submit/>

Following receipt of the application SWsoft will:

1. Send an email confirming receipt of the plug-in and asking for any further information.
2. Administer the certification process.
3. Notify the applicant of deficiencies (if any) or acceptance of the plug-in.

Following the acceptance of the plug-in, SWsoft will post a link to a web page where HSPcomplete customers can go to pay for (if applicable) and download the plug-in. This page must also provide support information for customers who have questions regarding the installation and maintenance of the plug-in.

---

## 2 PREREQUISITES

---

To apply for certification, the plug-in applicant should provide SWsoft with the following:

1. plug-in source code (as SRPM or tar archive with included SPEC file);
2. gateway test account (if plug-in uses any third party gateways);
3. documentation;
4. point of contact on gateway side;
5. filled certification procedure checklist (this document)

---

### 3 CERTIFICATION PROCEDURE CHECKLIST

---

#### 3.1 COMMON REQUIREMENTS

The following elements are common requirements amongst all HSPcomplete plug-ins.

##### 3.1.1 PLUG-IN INSTALLATION

	Item	Value or Status
<input type="checkbox"/>	Plug-in is packaged as an RPM	
<input type="checkbox"/>	Plug-in RPM package has correct set of dependencies	
<input type="checkbox"/>	Plug-in can be built into an RPM from the source code	
<input type="checkbox"/>	Plug-in installs correctly, using HSPcomplete upgrade system	

##### 3.1.2 PLUG-IN UNINSTALLATION

	Item	Value or Status
<input type="checkbox"/>	Plug-in can be uninstalled by rpm -e command	
<input type="checkbox"/>	Plug-in retains previous configuration when it is reinstalled	

#### 3.2 PAYMENT PLUG-IN REQUIREMENTS

The following elements are requirements of all Payment plug-ins.

##### 3.2.1 PRESENTATION CLASS CHECKS

	Item	Value or Status
<input type="checkbox"/>	Presentation class is in the following namespace: <b>HSPC::Plugin::PP::</b> It is composed of two parts: prefix OP or BT and real plug-in ID separated from each other by '_' sign. Example: OP_Anet (for Online Payment Authorize.net plug-in)	
<input type="checkbox"/>	There are no HSPcomplete internal objects or classes used except for <b>HSPC::PluginToolkit::General</b> or <b>HSPC::PluginToolkit::PP</b>	
<input type="checkbox"/>	The following mandatory methods are defined: <ul style="list-style-type: none"><li><input type="checkbox"/> view_form - returns string with HTML code that represents the view form of the plug-in configuration</li><li><input type="checkbox"/> edit_form - returns string with HTML code that represents edit/add form of the plug-in configuration</li><li><input type="checkbox"/> collect_data - returns a reference to a perl structure (reference to a hash is preferable)</li><li><input type="checkbox"/> get_help_page - returns a string with help page formatted as HTML</li></ul>	

3.2.2 MIDDLE TIER CLASS CHECKS

	Item	Value or Status
<input type="checkbox"/>	<p>Middle Tier (MT) class is in the following namespace:  <b>HSPC::MT::Plugin::PP::</b></p> <p>It should be composed from two parts: prefix OP or BT and real plug-in ID separated from each other by '_' sign.</p> <p>Example: OP_Anet (for Online Payment Authorize.net plug-in).</p>	
<input type="checkbox"/>	<p>There are no HSPcomplete internal objects or classes used except <b>HSPC::PluginToolkit::General</b> or <b>HSPC::PluginToolkit::PP</b></p>	
<input type="checkbox"/>	<p>The following mandatory methods are defined:</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> get_title - returns localized string</li> <li><input type="checkbox"/> get_supported_payment_method_types - returns a reference to an array of payment methods IDs</li> <li><input type="checkbox"/> get_currencies_supported - returns a reference to an array of currency ISO codes</li> <li><input type="checkbox"/> Either process_sale or process_preauthorize and process_capture methods - returns a reference to a hash with the following keys: <ul style="list-style-type: none"> <li>o STATUS (mandatory); Possible values are: APPROVED, FRAUD, DECLINED, PENDING, REDIRECT;</li> <li>o TEXT (mandatory) reference to a hash with two keys: <b>customer_message</b> – contains text that will be shown to customer as status message, and <b>vendor_message</b> – contains text that will be shown to vendor;</li> <li>o TRANSACTION_DETAILS (optional);</li> <li>o NEXT_TRANSACTION_GAP (optional);</li> <li>o ACTION (optional);</li> <li>o CUSTOM_RESPONSE (optional);</li> <li>o REDIRECT_HASH (optional);</li> </ul> </li> </ul>	
<input type="checkbox"/>	<p>List of supported payment method types contains known types provided by existing payment method plug-ins. (Example: hspc-plugin-pm-op-ccard). Otherwise there should be a new payment method plug-in.</p>	

3.2.3 CONFIGURATION CHECKS

	Item	Value or Status
<input type="checkbox"/>	<p>The installed plug-in is available for selection during creation of a new configuration and the name of plug-in is in human-readable format.</p>	
<input type="checkbox"/>	<p>Create a new configuration for the plug-in using the provided test account information.</p>	
<input type="checkbox"/>	<p>Save configuration and review it at the view page of plug-in configuration. All passwords should be hidden.</p>	

	Item	Value or Status
<input type="checkbox"/>	Proceed to the edit configuration page and check if all previously entered fields are pre-filled.	
<input type="checkbox"/>	Amend configuration parameters in order to make them incompatible with plug-in operation and save configuration. An appropriate error should appear.	
<input type="checkbox"/>	Check that at least English content for help pages at all 3 stages of plug-in configuration is available.	

### 3.2.4 RUNTIME CHECKS

	Item	Value or Status
<input type="checkbox"/>	If plug-in supports the Sale type of transaction, configure plug-in for it and made a transaction using test account at the payment gateway. Check that transaction is successful and all transaction information in the transaction log details page displays correctly.	
<input type="checkbox"/>	If plug-in supports the Credit type of transaction, make this transaction by refunding the Online Payment created as result of previous transaction. Check that the transaction was successful and all transaction information in the transaction log details page displays correctly.	
<input type="checkbox"/>	If plug-in supports the Preauthorize/Capture types of transaction, make these transactions using test account at the payment gateway. Check that the transaction was successful and all transaction information in the transaction log details page displays correctly.	
<input type="checkbox"/>	If plug-in support the Preauthorize Void/Capture Void types of transaction, make this transaction by refunding the Online Payment created as result of previous transaction. Check that the transaction was successful and all transaction information in the transaction log details page displays correctly.	

### 3.3 DOMAIN PLUG-IN REQUIREMENTS

The following are requirements of all Domain plug-ins.

#### 3.3.1 PRESENTATION CLASS CHECKS

	Item	Value or Status
<input type="checkbox"/>	Presentation class is in the following namespace: <b>HSPC::Plugin::DM::</b> . Class name should be real plug-in ID. Example: eNom.	
<input type="checkbox"/>	There are no HSPcomplete internal classes or objects used except ones that are placed in <b>HSPC::PluginToolkit::</b> and <b>HSPC::PluginToolkit::DM::</b> namespaces.	

### 3.3.2 MIDDLE TIER CLASS CHECKS

	Item	Value or Status
<input type="checkbox"/>	MT class is in the following namespace: <b>HSPC::MT::Plugin::DM::</b> <b>Class</b> name should be real plug-in ID. Example: eNom.	
<input type="checkbox"/>	Mandatory method <b>register_domain</b> is defined	

### 3.3.3 CONFIGURATION CHECKS

	Item	Value or Status
<input type="checkbox"/>	The installed plug-in is available for selection during creation of new configuration and the name of plug-in is in human-readable format.	
<input type="checkbox"/>	Create a new configuration for the plug-in using the provided test account information.	
<input type="checkbox"/>	If plug-in supports viewing configuration settings then review it at view page of plug-in configuration. All passwords should be hidden.	
<input type="checkbox"/>	If plug-in supports editing configuration settings then proceed to the edit configuration page and check if all previously entered fields are pre-filled.	
<input type="checkbox"/>	Amend configuration parameters in order to make them incompatible with plug-in operation and save configuration. An appropriate error should appear.	
<input type="checkbox"/>	Amend configuration parameters in order to make them compatible with plug-in operation and save configuration. Check that changes were saved.	
<input type="checkbox"/>	Proceed to plug-in TLD configuration page and add all supported TLDs. Check if parameters of all supported features are available for configuration for added TLDs.	
<input type="checkbox"/>	Check that at least English content for help pages at all stages of plug-in configuration is available.	
<input type="checkbox"/>	<p>Check that contact replacement feature is implemented correctly (if supported by plug-in):</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Check if all expected contact types are available on plug-in TLD configuration page.</li> <li><input type="checkbox"/> Check if all expected fields are available and prefilled if it's necessary on the contact information form.</li> <li><input type="checkbox"/> Amend contact fields values in order to make them incorrect in accordance with validation rules and try to save contact. An appropriate error should appear.</li> <li><input type="checkbox"/> Amend contact fields values in order to make them correct in accordance with validation rules and try to save contact. Check that changes were saved.</li> </ul>	

3.3.4 RUNTIME CHECKS

	Item	Value or Status
<input type="checkbox"/>	<p>If plug-in supports Name Server Registration feature then check that you can register at least 2 Name Servers. Check that you can register at least 2 domains for each supported TLD.</p>	
<input type="checkbox"/>	<p>One domain should be registered for corporate account and other one for personal account. Check contact information forms in the same way as for “Contact Replacement” feature at plug-in configuration checks. Also check Domain Extended Data form in the same manner.</p>	
<input type="checkbox"/>	<p>If plug-in supports Domain Renewal feature then check that you can renew at least 1 registered domain for each supported TLD.</p>	
<input type="checkbox"/>	<p>If plug-in supports Domain Transfer feature then check that you can transfer at least 1 domain for each supported TLD. Check contact information forms in the same way as for “Contact Replacement” feature at plug-in configuration checks. Also check Domain Extended Data form in the same manner.</p>	
<input type="checkbox"/>	<p>If plug-in supports Domain Termination feature then check that you can terminate at least 1 registered domain for each supported TLD.</p>	