iSCert Hub

**Manual de Generación de Instaladores**

|  |  |  |
| --- | --- | --- |

Versión: 1.0

Fecha: 08/05/2019

**Historial** de Revisiones

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 08/05/2019 | 1.0 | Creación del documento | Mauricio Irace |
|  |  |  |  |
|  |  |  |  |

**Índice**

[**1. Introducción 2**](#_gjdgxs)

[**2. tutorial Paso a paso 3**](#_30j0zll)

[**3. Estructura de carpetas 4**](#_1fob9te)

[**4. Creando un nuevo instalador para un nuevo plugin 7**](#_3znysh7)

# Introducción

Este Manual consiste

# INTRODUCCIón

Supongamos que hemos creado un plugin **jaime**, que al ser invocado, cambia el fondo del cliente por una foto aleatoria de Jaime Roos. Nuestro flamente plugin a pasado ya por fase alfa, beta testing,ha sido corroborado por la contraparte de cliente, y esta listo para salir en producción. Lamentablemente, Mauri es un traidor, y se fue, pero en un atisbo de bondad y conciencia, ha dejado este tutorial para explicarnos como generar un instalador para el iSCert Hub. Pero antes de saber como hacer esto, es necesario revisar la estructura de carpetas utilizada por el generador de instaladores.

# Estructura de carpetas

La raíz de la estructura se encuentra en: *Google Drive/iSCert Hub/installer* y es la siguiente:

Aunque puede parecer complicada al principio, y en efecto, tiene bastante contenido. Sin embargo, pronto se ira familiarizando con el ambiente. A continuación se detalla los elementos de la misma. No se preocupe sino se acuerdo de todo en el momento, porque trabajando sobre la misma irá entendiendo

* **common/** Aquí se encuentra el script base, que contiene los archivos *“core”* que son comunes a todos los instaladores.
* **common/cert/** Dependiendo de la legislación correspondiente a la región del cliente, puede ser necesario utilizar distintos certificados SSL para los Web Sockets que levanta el Hub. Mientras que isa.iss indica que hay que utilizar los certificados de isa, por cuestiones legales en el Ministerio de Interior paraguayo, existe py.iss, que contiene certificados válidos para el mismo.
* **ClientN/** Cada cliente tiene un script spec.iss, que contiene los archivos específicos a cada uno de los mismos. En general, estos archivos referencian al script base, al certificado que usen, y detallan los plugins que requieren.
* **ext/** Contiene los archivos y scripts independientes de Inno Setup utilizados por el mismo para generar los instaladores.
* **ext/bin/** Archivos binarios y comprimidos del wrapper del manager (archivo .exe que ejecuta el jar del manager) y del jar del servidor de websockets. Se posee respaldos de versiones alternativas (Por ejemplo, con sistema de licencia real o simulado. O versiones anteriores). Si ha de generar un hub con una de estas versiones, se ha de remplazar el contenido (previo respaldo del mismo) de la carpeta bin con el de alguna de sus subcarpetas. Este proceso no se ha automatizado todavía.
* ***ext/build/*** Scripts .bat que se encargan de simplificar algunos pasos tediosos de la compilación de los instaladores. Cuando se desee generar un instalador, en generar solo basta con ejecutar el script correcto.
* **ext/cert ext/jks** Certificados varios de ssl y claves privadas, respectivamente.
* **ext/config/** Configuración encriptada de los plugins y puertos. A su vez, contiene la configuración plana de los dll de los tokens.
* **ext/dist/**  Aquí se guardan los instaladores cuando son generados. Se recomienda mover al medio de distribución con el cliente.
* **ext/lib/** diretorio donde se encuentran los plugins
* **ext/l4j** archivos del wrapper del manager, basado en launch4j
* **ext/fox/ ext/foxpy S**cripts de instalación de certificados para el almacén de Firefox.
* **ext/jre/** Jre que incluido pro el Hub, para evitar incompatibilidades de versiones.
* **ext/license** Certificados para licencias, y licencias ya generadas.
* **ext/plane** Archivos de configuración en plano. Hay uno por cliente, puesto que los mismo determinan los plugins que a los que tendrá accesso. (Se mantiene “plane” (raíces de ax + by + cz + d), en vez “plain”(Objeto no alterado por primitivas criptográficas) por un error ortográfico de legado.)
* **ext/scripts** scripts varios que se instalan con la distribución
* **ext/system** Bibliotecas .dll usadas por algunos plugins que requieren acceso de bajo nivel.
* ***ext/build.bat****nombre\_carpeta\_cliente* Compila el hub para un cliente específico.

# Creando un nuevo instalador para un nuevo plugin

El primer paso, es renombrar el plugin de *nombre.jar*, a *plugin.nombre.isc*, por convención.

El segundo, consiste en moverlo a la carpeta ***ext/*lib/.**

Luego, copiar la carpeta **template** de la raíz (de la carpeta installer), y cambiarle el nombre por el del cliente o el que quiera darle a la distribución. De ahora en más, llamaremos $*name* a este valor

Dentro de esta nueva carpeta, modificar el archivo *spec.iss:*

* Cambiar el OutputBaseFilename (nombre del instalador) por el deseado.
* En la sección files, borrar el plugin de ejemplo (pues no existe) y cambiar por los que desee incluir. Por ejemplo, cambiar *plugin.template.isc* por *plugin.sign-p7.isc*

Crear **ext/plane/plugins/$name** como copia de la carpeta template del mismo directorio. Reemplazar el contenido de service.yml con la configuración del plugin nuevo, así como los otros plugins que quiera incluir.

Listo, ahora debería poder crear un instalador ejecutando **build.bat** *$name.* Opcionalmente puede crear un archivo en build bat que llame a este comando, para generar con un simple doble click. El instalador se generar en la carpeta **ext/dist/**