

SISTEMA DIGITAL PARA EL MANEJO DE INFORMACIÓN DE ORGANISMOS VIVOS MODIFICADOS

HAPI – Hermes Administrador de Procesos e Información

MANUAL TECNICO
Versión 10.0

www.ovm.go.cr

Este Sistema fue desarrollado con el aporte del Programa de las Naciones Unidas para el Medio Ambiente (UNEP, *por sus siglas en inglés*) y el Fondo para el Medio Ambiente Mundial (GEF, *por sus siglas en inglés*), bajo el marco del Proyecto No. GLF/2328-2716-4B61 "Implementación de un Marco Nacional de Bioseguridad para Costa Rica"

Se le agradece a Hermes Soluciones de Internet S.A. por el desarrollo de la herramienta. San José, Costa Rica www.hermes-soft.com

Identificación del documento

Datos del Documento

Nombre de archivo:	manual tecnico HAPI.docx
Fecha de creación:	21 setiembre 2009
Última modificación:	25 abril 2013

Historia de Revisión del Documento

Fecha	Versión	Información de los Cambios
21 agosto 2009	1.0	Versión inicial: requisitos de instalación
30 agosto 2009	2.0	Actualización de requisitos Guía de instalación
13 setiembre 2009	3.0	Guía de integración
13 octubre 2009	4.0	Cambios por firma digital agente electrónico
18 octubre 2009	5.0	Revisión general Instrucciones para instalar certificado de agente electrónico
6 diciembre 2009	6.0	Parámetros y procedimientos para HAPI
2 marzo 2011	7.0	Instrucciones para configurar proxy .Net Framework 4.0
20 setiembre 2011	8.0	Manuales técnicos de HAPI y firma digital se unifican Cambios por ticket #242
26 octubre 2011	8.1	Integración con AlwaysUp (ticket #381)
15 diciembre 2011	8.2	Configuración de los clientes
24 mayo 2012	9.0	Actualización del manual para la versión 4.13 de HAPI
25 abril 2013	10.0	Actualización del manual de acuerdo al ticket 1139

TABLA DE CONTENIDOS

INTRODUCCIÓN 8

REQUERIMIENTOS DE INSTALACIÓN 8

REQUERIMIENTOS DEL SERVIDOR WEB 9
Parámetros de desempeño del servidor web 9
REQUERIMIENTOS DEL SERVIDOR DE APLICACIONES 11
Parámetros de desempeño del servidor de aplicaciones 14
REQUERIMIENTOS DEL SERVIDOR DE BASE DE DATOS 14

GUÍA DE INSTALACIÓN 16

CREACIÓN DE BASE DE DATOS 16
INSTALACIÓN DE SERVIDOR DE APLICACIONES 20
Copiar los archivos del servidor de aplicaciones 20
Instalar el componente cliente de firma digital 20
Instalar el runtime de Visual J 20
Registrar el componente servidor de firma digital 20
Registrar itextsharp.dll en GAC 21
Registrar ICSharpCode.SharpZipLib.dll en GAC 21
Crear colas de servidor de aplicaciones 21
Instalación y configuración de aplicaciones auxiliares 23
Instalación de memcached 23
Instalación de Microsft Enterprise Library 5 24
Instalación de ImageMagick 6.7.5 para Windows 25
Configuración de Ghostscript 9.02 para Windows 25
Instalación de tesseract 3.01 para Windows 25
Instalación del solr 25
Ajustar parámetros de DSAppServer.exe.config 27
Personalizar diseño grafico de las notificaciones 30
Ajustar rutas de HafFormToXslFo.xsl 30
Conexión a base de datos 30
Asignar puerto usado por servidor de aplicaciones 31
Instalar y configurar el servicio de HAPI 31
Ajustar parámetros del servidor de aplicaciones (DSAppServer.exe.config) 32
Parámetros del servicio de firma digital de agente electrónico 33
Configuración del proxy 34
Instalación del servidor de aplicaciones con AlwaysUp 34
INSTALACIÓN DE APLICACIÓN WEB 41
Copiar archivos de la aplicación web 41
Registrar aplicación web en el IIS 42
Definir documento por defecto 42

Copiar dlls de ReportViewer *42*
Instalar el runtime de Visual J *42*
Registrar el componente servidor de firma digital. *43*
Actualizar dirección del servidor de aplicaciones *43*
Ajustar rutas de web.config *43*
Poner ruta de webservice de validación *44*
Instalar el componente cliente de firma digital. *44*
Instalación de Microsft Enterprise Library 5 ..... *44*
Configurar sección mailSettings *44*
Copiar dlls de exportación a pdf en System32 ..... *45*
Creación de usuario administrador *45*
Creación de horario "Normal" *45*
Configuración de la autoridad certificadora *45*
Registro de extensión .fhp en IIS 7 ..... *46*
Activar compresión en el IIS para HermesDSApp *46*

GUÍA DE INTEGRACIÓN CON ASP 48

Archivo HermesDS.inc *48*
Integración del componente de firma digital *50*
Validación de la firma digital *53*
Firma digital de agente electrónico *54*
Generación del comprobante 55
Formato de xml para comprobante 56
Descarga del comprobante mediante llamado a función 57
Descarga del comprobante mediante DescargarComprobante.asp 58
Instalación de un certificado de agente electrónico *58*
Instalar certificado en el store de certificados de la computadora local 58
CONFIGURACIÓN DEL FIRMADOR 59
Configurar la dirección del webservice de validación *59*
Configurar la dirección de instalación del firmador *59*
CONFIGURACIÓN DE LAS COMPUTADORAS DE LOS USUARIOS 62
Agregar HAPI a los sitios de confianza *62*
Deshabilitar bloqueadores de ventanas emergentes (popups) *62*

Tabla de ilustraciones

Ilustración 1 Diagrama de emplazamiento del sistema de firma digital 8
Ilustración 2 Instalación del Message Queue Server 13
Ilustración 3 Crear cola privada "ColaHaf" 22
Ilustración 4 Permisos de la cola "ColaHaf" para "Network Service" 23
Ilustración 5 Código ASP integrado con firma digital 51
Ilustración 6 Referencia generada a digsig.js 51
Ilustración 7 Invocar función para firmar digitalmente 52
Ilustración 8 Campo generado para contener firma digital 52
Ilustración 9 Función generada para firmar digitalmente 53
Ilustración 10 Referencia generada a Activex de firma digital 53
Ilustración 11 Código ASP para validar una firma digital 54
Ilustración 12 Listado de código para generar comprobante 55
Ilustración 13 Fragmento de comprobante generado con comprobante-fo.xsl 56
Ilustración 14 Formato XML esperado por comprobante-fo.xsl 56
Ilustración 15 Ejemplo de uso de función GetPDF 57
Ilustración 16 Ejemplo de invocación a página DescargarComprobante.asp 58

Introducción

Este documento brinda instrucciones para:

- Instalar y configurar la aplicación web de HAPI
- Instalar y configurar el servidor de aplicaciones de HAPI
- Crear e inicializar la base de datos de HAPI
- Implementar la firma digital de formularios web en una página ASP

Requerimientos de Instalación

El servicio de firma digital de Hermes se puede instalar en una configuración de tres capas: presentación, lógica del negocio y base de datos. Cada capa puede emplazarse en un servidor dedicado o en varios servidores en un ambiente de balanceo de carga.

Las siguientes secciones describen los requisitos para cada uno de los servidores involucrados en cada capa.

El siguiente diagrama muestra los diferentes componentes de la solución de firma digital de Hermes y su emplazamiento en los diferentes servidores.

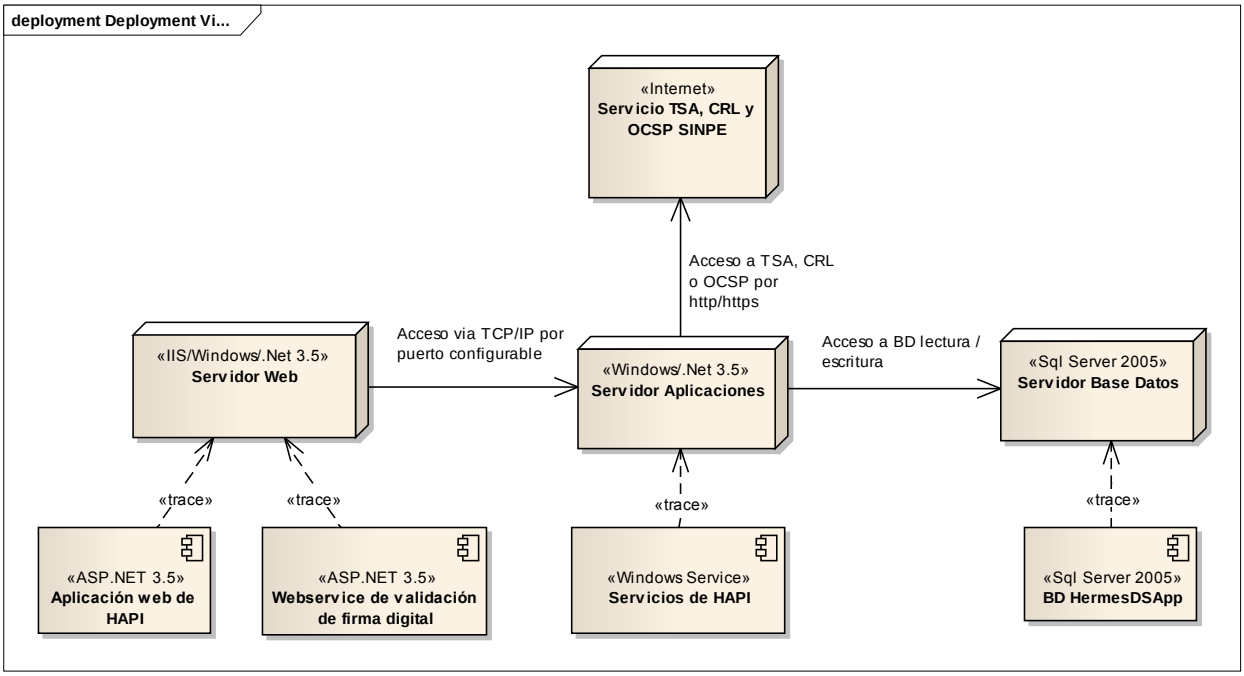


Ilustración 1 Diagrama de emplazamiento del sistema de firma digital

Nótese que aunque la aplicación puede distribuirse en tres capas de servidores también es posible instalar en un solo servidor o en dos servidores todos los componentes.

Requerimientos del servidor web

El servidor web hospedara la aplicación web de HAPI, desarrollada en ASP.NET 4.0. La aplicación web se comunicará con el servidor de aplicaciones mediante un puerto TCP/IP el cual debe ser definido por el cliente. Esta comunicación puede ser directa o puede ser por medio de un NLB. Para la aplicación web se requiere:

- Windows 2003 SP2 o superior
- IIS 6 o superior
- Servicio ASP.NET Session State
- MVC 2 instalado
- .NET Framework 4.0
  El instalador se encuentra en la ruta \fdg\dotNetFx40_Full_x86_x64.exe del CD de instalación
- Acceso al servidor de aplicaciones por medio de un puerto definido por el cliente
- Microsoft Report Viewer Redistributable 2010
  El instalador se encuentra en la ruta \fdg\ReportViewer.exe del CD de instalación
- *Se requieren 16 GB RAM*

Nótese que desde el servidor web no se requiere acceso directo a la base de datos ya que todo el acceso a la base de datos se realiza mediante el servidor de aplicaciones.

Parámetros de desempeño del servidor web

Se recomiendan los siguientes ajustes a nivel de la configuración general de los servidores web para aumentar el desempeño de la aplicación.

En los archivos

- C:\Windows\Microsoft.NET\Framework\v2.0.50727\CONFIG\Machine.config
- C:\Windows\Microsoft.NET\Framework64\v2.0.50727\CONFIG\Machine.config

Realizar los siguientes ajustes indicados en negrita:

<system.web>

…

<processModel maxWorkerThreads="150" maxIoThreads="150" minWorkerThreads="50" minIoThreads="50"/>

…

</system.web>

<system.net>

…

<connectionManagement>

<add address="*" maxconnection="150"/>

</connectionManagement>

…

</system.net>

En los archivos

- C:\Windows\Microsoft.NET\Framework\v2.0.50727\aspnet.config
- C:\Windows\Microsoft.NET\Framework64\v2.0.50727\aspnet.config

Realizar los siguientes ajustes indicados en negrita

<system.web>

…

<processModel maxWorkerThreads="150" maxIoThreads="150" minWorkerThreads="50" minIoThreads="50"/>

<applicationPool maxConcurrentRequestsPerCPU="5000" maxConcurrentThreadsPerCPU="0" requestQueueLimit="5000"/>

…

</system.web>

<system.net>

…

<connectionManagement>

<add address="*" maxconnection="150"/>

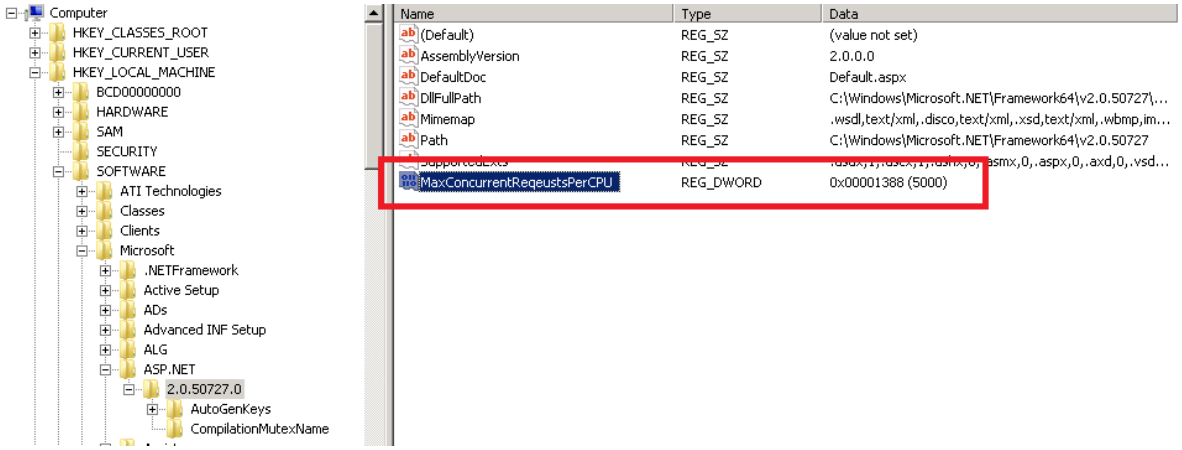
</connectionManagement>

…

</system.net>

Crear la siguiente entrada de tipo DWORD en el registry de los servidores web con valor 5000

Revisar
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ASP.NET\2.0.50727.0\MaxConcurrentRequestsPerCPU



Requerimientos del servidor de aplicaciones

El servidor de aplicaciones hospedará los servicios de HAPI los cuales implementan toda la lógica de validación de firma digital, actualización de la bitácora, ejecución y control de los flujos de trabajo. Los servicios de HAPI corren dentro de un servicio de Windows el cual escucha por

un puerto TCP/IP las solicitudes realizadas desde la aplicación web de HAPI. Este puerto TCP/IP debe ser definido por el cliente.

El servidor de aplicaciones se conecta a la base de datos para realizar consultas o actualizar la bitácora firmada digitalmente.y la base de datos de HAPI. La conexión a la base de datos puede autenticarse utilizando autenticación integrada de Windows o un usuario de Sql Server 2005.

Para el servicio Windows de HAPI se requiere:

- Windows 2003 SP2 o superior
- MSDTC Activo
- .NET Framework 4.0
- Java SE RT 1.6 o superior para Windows (requerido para solr)
  El instalador de java se encuentra en \fdg\jre-6u32-windows-i586.exe
- 16 GB RAM
- 200 GB disponibles en el disco duro
- Instalación basica de MS Message Queuing. No se requiere integración con active directory. Para determinar si este componente está instalado debe verificarse que existe el nodo “Message Queueing” debajo del nodo “Services and Applications” de la consola “Computer Management” (click Start | Control Panel , doble clic en icono “Administrative tools” y doble clic en icono “Computer Management”)

  Para instalar el Message Queue Server se deben seguir los siguientes pasos:

  Clic en Start | Control Panel, doble clic en icono “Programs and features” y clic en el enlace “Turn Windows features on or off”

  Seleccionar Microsoft Message Queue (MSMQ) Server y seleccionar los componentes que se muestran a continuación (el único que no es requerido seleccionar es “MSMQ Active Directory Domain Services Integration”.

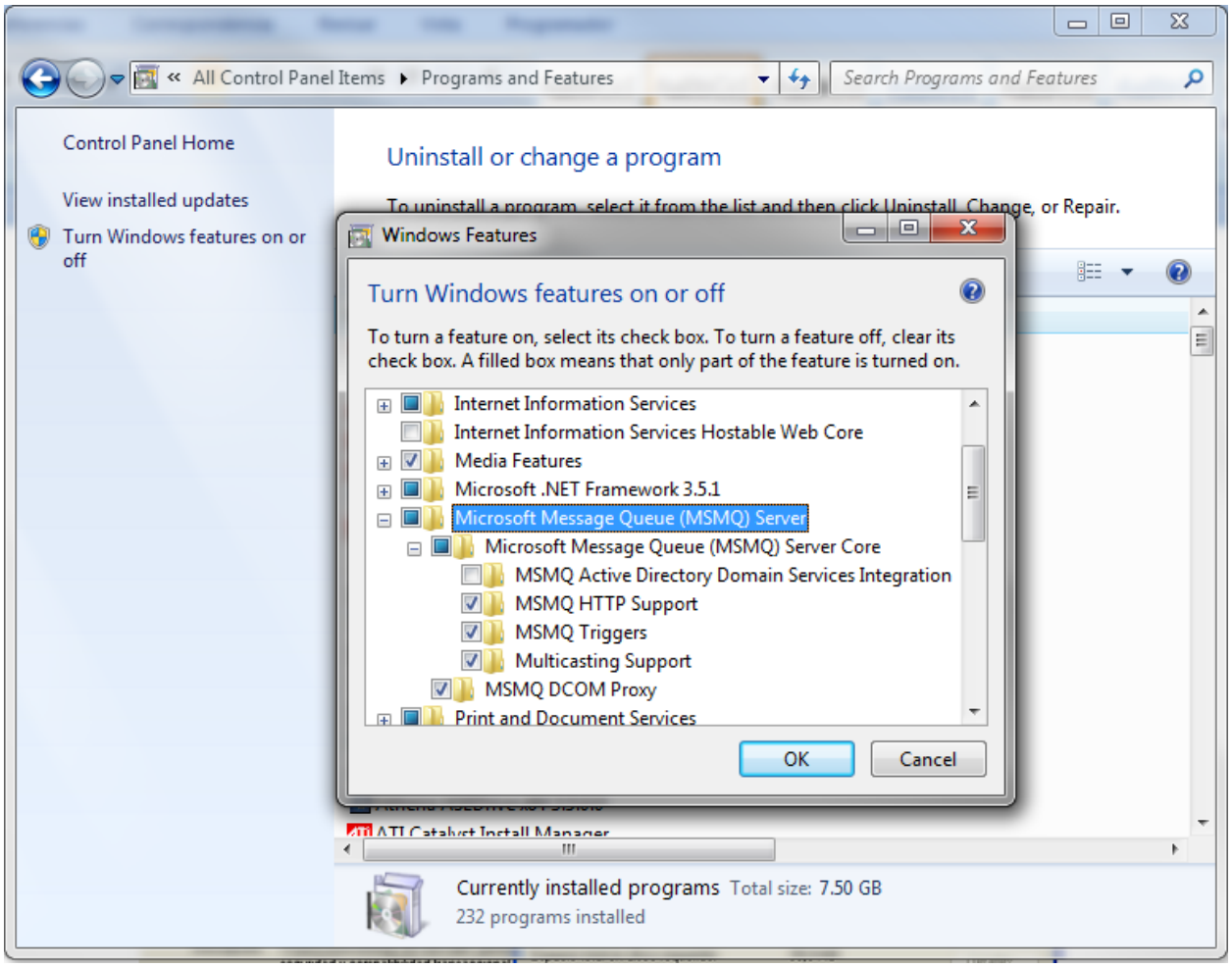


Ilustración 2 Instalación del Message Queue Server

- Si la autenticación a la base de datos se realiza utilizando autenticación integrada de Windows entonces el servicio requiere correr con un usuario con permiso de lectura y escritura en la base de datos de HAPI (HermesDSApp).
- Si la autenticación a la base de datos se realiza utilizando un usuario de SQL Server entonces el servicio debe correrse con el usuario “Network Service” o algún otro usuario con acceso a la red.
- Acceso por Internet, http (puerto 80) y https (puerto 443), a cualquier URL en los dominios:
    - fdi.sinpe.fi.cr
    - firmadigital.go.cr

- .NET Framework 4.0
  El instalador se encuentra en la ruta \fdg\dotNetFx40_Full_x86_x64.exe del CD de instalación
- Opcionalmente para instalar el servicio con AlwaysUp se debe instalar dicho software en cada servidor de aplicaciones

Parámetros de desempeño del servidor de aplicaciones

Con el fin de aumentar el desempeño de la aplicación es necesario hacer algunos ajustes a nivel de los archivos de configuración de .NET

En los siguientes archivos

- C:\Windows\Microsoft.NET\Framework\v2.0.50727\CONFIG\Machine.config
- C:\Windows\Microsoft.NET\Framework64\v2.0.50727\CONFIG\Machine.config

realizar los siguientes cambios:

<system.web>

…

<processModel maxWorkerThreads="500" maxIoThreads="500" minWorkerThreads="50" minIoThreads="50"/>

…

</system.web>

Requerimientos del servidor de base de datos

La base de datos de HAPI (Sql Server 2005 o superior) es accedida por el servidor de aplicaciones para realizar consultas y para insertar o actualizar datos. Requisitos del servidor de base de datos:

- SQL Server 2005 o superior
- MSDTC Activo
- 16 GB RAM
- 500 GB disponibles en disco duro
- Una base de datos nueva llamada HermesDSApp
- Si se va a utilizar autenticación integrada de windows para conectarse a la base de datos entonces se requiere que el usuario de Windows sea registrado en la base de datos HermesDSApp y que tenga permisos de lectura y escritura sobre las tablas de

HermesDSApp y permisos de ejecución de procedimientos almacenados de la base de datos HermesDSApp.

- Si se va a utilizar autenticación de Sql Server para conectarse a la base de datos HermesDSApp entonces se requiere que el usuario de Sql Server sea registrado en la base de datos HermesDSApp y que tenga permisos de lectura y escritura sobre las tablas de la base de datos HermesDSApp y permisos de ejecución de procedimientos almacenados de la base de datos HermesDSApp.

Guía de instalación

Creación de base de datos

Crear una base de datos llamada HermesDSApp en Sql Server 2005. Si se utiliza otro nombre para la base de datos entonces la configuración del servidor de aplicaciones debe actualizarse para reflejar este cambio.

Ejecutar los siguientes scripts en la base de datos HermesDSApp, en el mismo orden del documento:

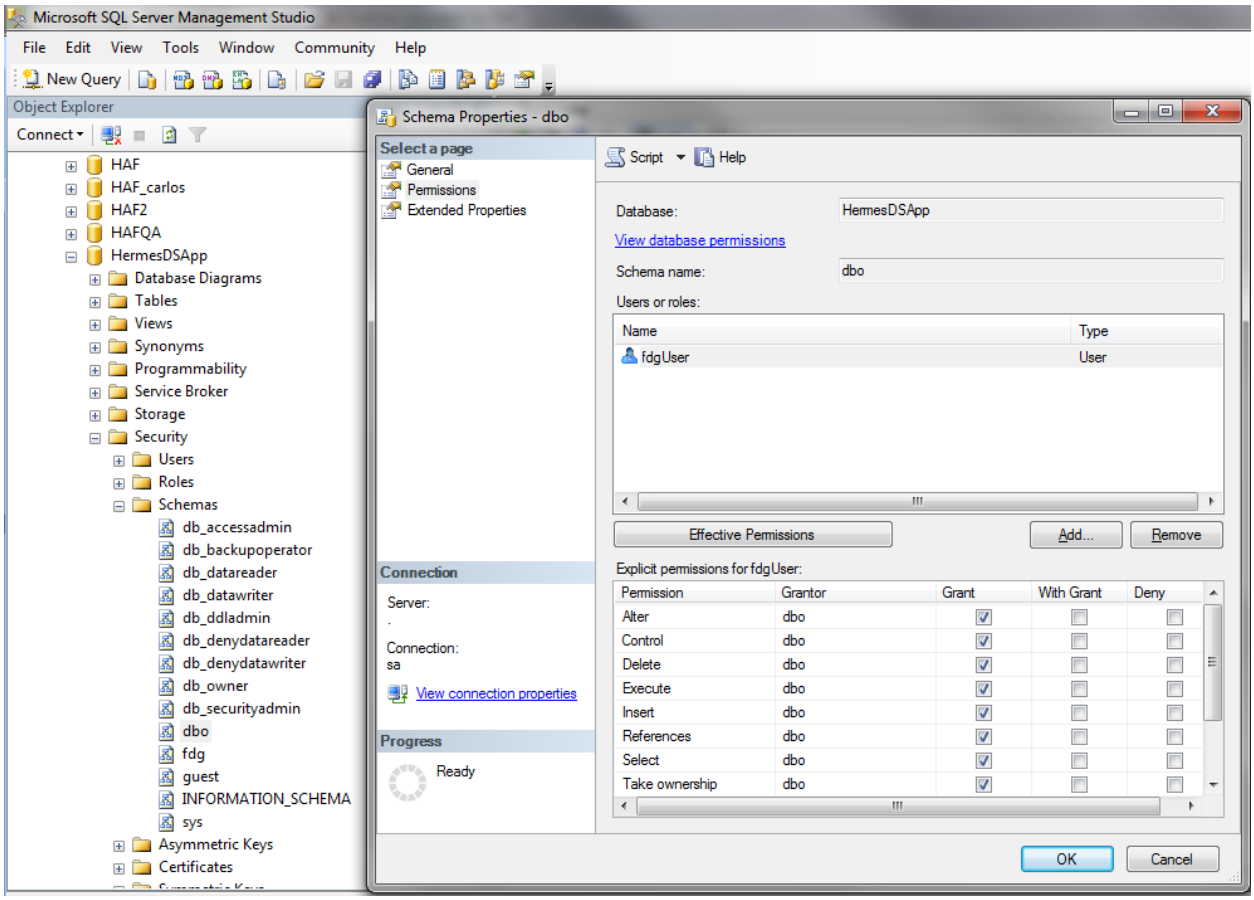
Script	Propósito
Módulo de autenticación	
aut_tablas.sql	Crea las tablas para el módulo de autenticación y control de acceso
aut_sps.sql	Crea los procedimientos almacenados para el módulo de autenticación y control de acceso
aut_inicializar.sql	Incluye algunos registros en tablas lookup
aut_vistas.sql	Vistas
Módulo de flujos de trabajo	
haf_tablas.sql	Crea las tablas para el módulo de flujos
haf_vistas.sql haf_VistaSolicitud.sql	Crea vistas para el módulo de flujos
haf_inicializar.sql	Incluye algunos registros en tablas lookup
haf_agregar_actor.nombre.sql	Varios triggers para sincronizar haf.Actor cuando cambia aut.Usuario
Ticket #725 Tabla haf.CatalogoGeneral.sql	Script para crear e inicializar la tabla haf.CatalogoGeneral

Ticket #1066 Códigos UNESCO.sql	Script para cargar los códigos UNESCO en la tabla haf.CatalogoGeneral
Módulo de firma digital	
Tablas.sql	Tablas principales del módulo de firma digital
Indices.sql	Índices necesarios para mejor desempeño
funciones.sql	Funciones de base de datos para reportes
vistas.sql	Crea vistas para el módulo de firma digital
sps.sql	Crea procedimientos almacenados para el módulo de firma digital
inicializar.sql	Incluye algunos registros en tablas lookup
Módulo de repositorio	
docs_tablas.sql	Tablas del módulo del repositorio esquema (docs)
docs_funciones.sql	Funciones del módulo de repositorio esquema (docs)
trigger Repositorio.afterInsertRepositorio.sql	Script para crear trigger que carga reglas iniciales de un repositorio
Tablas de Windows Workflow Foundation	
SqlPersistenceService_Schema.sql	Esquema de tablas de WWF
SqlPersistenceService_Logic.sql	Procedimiento de tablas de WWF
Tracking_Schema.sql	Tablas para tracking de WWF
Tracking_Logic.sql	Procedimientos de tablas de tracking de WWF

Dichos scripts se encuentran en la ruta \sql scripts\ del CD de instalación

Los usuarios hafUser, autUser y fdgUser deben tener permisos para ejecutar, seleccionar, insertar, eliminar, actualizar, etc. sobre el esquema dbo de la base de datos HermesDSApp como se muestra en la siguiente imagen.

Se debe asignar permisos de update y select al usuario autUser sobre la tabla haf.Actor (Ticket #477 Autorizar Update a autUser en Actor.sql)



Además el usuario hafUser debe tener permiso para ejecutar, seleccionar, insertar, eliminar, y actualizar en los esquemas fdg y aut.

Finalmente actualizar la columna toCRLDPID de la tabla fdg.CRLDP como se muestra en la siguiente imagen:

CRLDPID	ObjectVersion	URL	UltimaRevision	toCRLDPID
1	2	http://fdi.sinpe.fi.cr/repositorio/CA%20SINPE%20-%20PERSONA%20FISICA.crl	2011-10-01 23:...	2
2	2	http://www.firmadigital.go.cr/repositorio/CA%20POLITICA%20PERSONA%20FISICA%20-%20COSTA%20RICA.crl	2011-10-01 23:...	6
3	1	http://127.0.0.1/ca/crl/2009.crl	NULL	NULL
4	1	http://fdi.sinpe.fi.cr/repositorio/CA%20SINPE%20-%20PERSONA%20FISICA+.crl	2011-10-01 23:...	NULL
6	1	http://www.firmadigital.go.cr/repositorio/CA%20RAIZ%20NACIONAL%20-%20COSTA%20RICA.crl	NULL	NULL
NULL	NULL	NULL	NULL	NULL

Instalación de servidor de aplicaciones

Los siguientes son los pasos para instalar y configurar los componentes del servidor de aplicaciones de HAPI:

Copiar los archivos del servidor de aplicaciones

- Crear el directorio “d:\fdg” en el servidor de aplicaciones y asignarle permisos full-control al usuario “Network Service” sobre ese directorio
- Copiar en el directorio “d:\fdg” las carpetas “Servidor”, “NetAppServer”, “Queries”, “dtos”, “Anexos”,. “Formularios”, “TrustedCerts” y “Cliente” que se encuentran en el directorio “fdg” del CD.
- Copiar en el directorio “d:\fdg” los archivos “digsigcom-localmachine.reg”, “DigSigServerCOM.dll”, “DigSigCOM.dll”, “vjredist.exe”, “IDigitalSignatureComponente-localmachine.reg”, “gacutil.exe” y “itextsharp.dll” que se encuentran en el directorio “fdg” del CD.
- Crear el directorio d:\temp y ponerle permisos de “full control” al usuario “Network Service”

Instalar el componente cliente de firma digital

- Desde el cmd ejecutar *regsvr32 d:\fdg\digsigcom.dll*
- Desde el Explorer ir al directorio d:\fdg y hacer doble clic en los archivos digsigcom-localmachine.reg y IDigitalSignatureComponente-localmachine.reg para cargarlos en el registro de Windows.

Instalar el runtime de Visual J

- Desde el Explorer de Windows ir al directorio d:\fdg y ejecutar el instalador vjredist.exe. Estas librerías son necesarias para las funciones de exportación de datos.

Registrar el componente servidor de firma digital

Para el siguiente paso se requiere ejecutar el cmd con el usuario administrador:

- regsvr32 d:\fdg\digsigservercom.dll.

Registrar itextsharp.dll en GAC

- desde el cmd ejecutar:
- d:
- cd d:\fdg
- gacutil /i d:\fdg\itextsharp.dll

Registrar ICSharpCode.SharpZipLib.dll en GAC

- desde el cmd ejecutar:
- d:
- cd d:\fdg
- gacutil /i d:\fdg\NetappServer\ICSharpCode.SharpZipLib.dll

Crear colas de servidor de aplicaciones

- Crear una cola privada transaccional llamada “firmas” en el servidor de aplicaciones. Si se utiliza otro nombre entonces se debe reflejar el mismo en el parámetro ColaFirmas del archivo DSAppServer.exe.config

- Crear una cola privada transaccional llamada “colahaf” en el servidor de aplicaciones. Si se utiliza otro nombre entonces se debe reflejar el mismo en el parámetro ColaHaf del archivo DSAppServer.exe.config

- Crear una cola privada transaccional llamada “IndexadorHAPI” en el servidor de aplicaciones. Si se utiliza otro nombre entonces se debe reflejar el mismo en el parámetro ColaIndexador del archivo DSAppServer.exe.config

- Instrucciones para crear una cola:

  En la consola de Administración de equipos (clic Start | Administrative Tools, doble clic icono Administrative Tools, doble clic Computer Management), expandir el nodo “Servicios y aplicaciones”, expandir el nodo “Message Queuing”, clic derecho en “Colas Privadas” y clic en “Nuevo | Cola privada” como se muestra en la siguiente imagen:

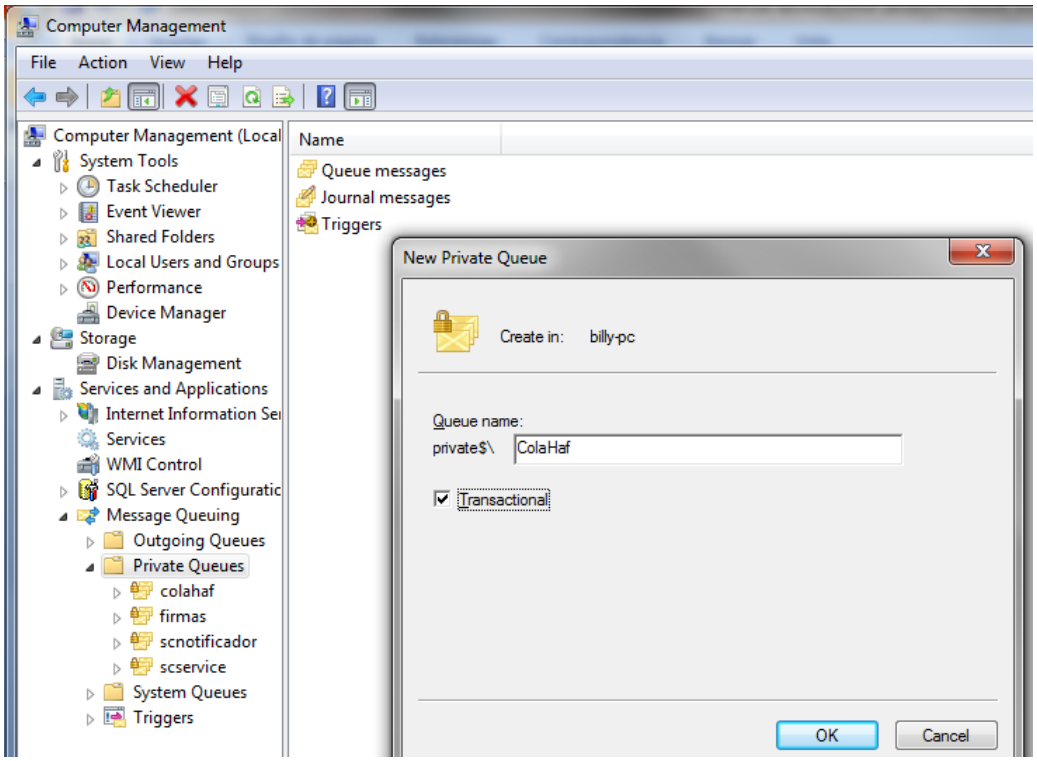


Ilustración 3 Crear cola privada "ColaHaf"

- Asignar permisos de full-control al usuario “Network Service” sobre la cola “colahaf” y “firmas”.

  Expandir nodo “Private Queues”, hacer clic derecho en la cola “colahaf”, clic en la opción “Properties” del menú de contexto, clic en la pestaña “Security” e incluir el usuario “Network Service” con “Full Control”

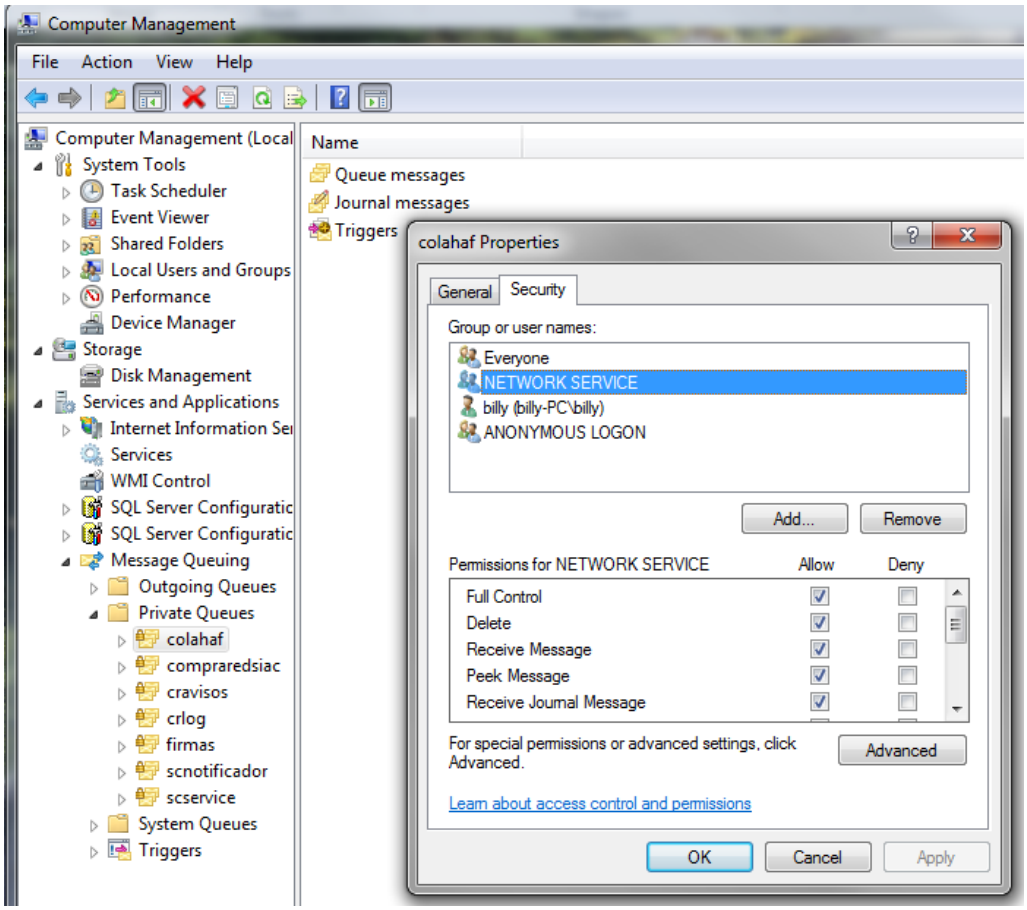


Ilustración 4 Permisos de la cola "ColaHaf" para "Network Service"

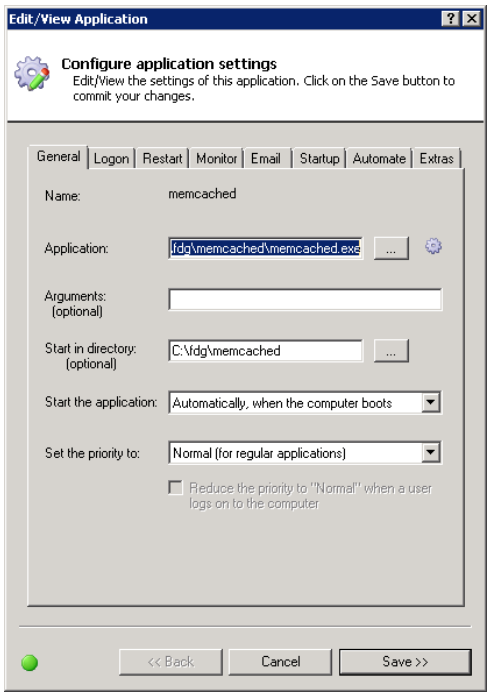
Instalación y configuración de aplicaciones auxiliares

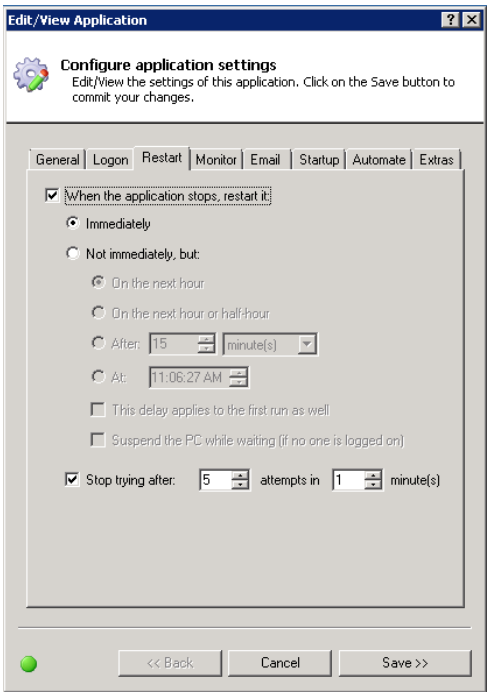
En esta sección se explica la instalación de diversas aplicaciones utilizadas por el servicio de HAPI para varias funciones auxiliares.

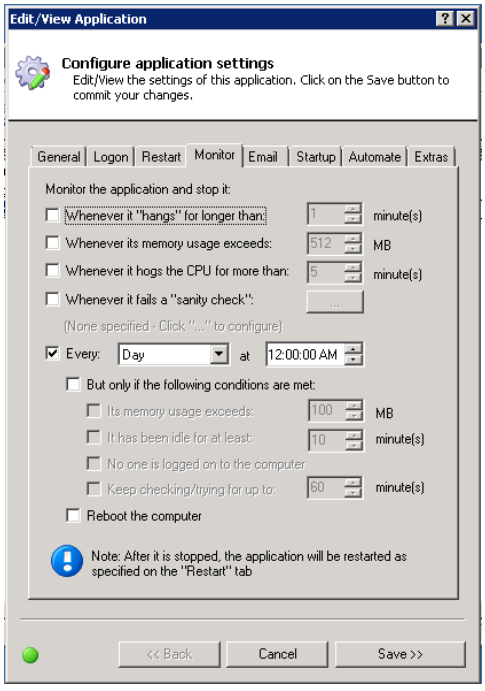
Instalación de memcached

Este software es utilizado por HAPI como un cache de nivel 2. Para instalar este software simplemente se debe copiar el directorio \fdg\memcached del CD de instalación en el directorio c:\ del servidor de aplicaciones.

Una vez instalador el memcached se debe ejecutar desde el Alwaysup con la siguiente configuración:







Instalación de Microsft Enterprise Library 5

Este software contiene varios componentes de Microsoft que son utilizados por HAPI. Para instalar este software se debe ejecutar el instalador que está en la ruta \fdg\Enterprise Library 5.0.msi del CD de instalación.

Instalación de ImageMagick 6.7.5 para Windows

Este software lo utiliza el servicio de HAPI para rotar y escalar imágenes. Ejecutar el instalar que se encuentra en la ruta \fdg\ImageMagick-6.7.5-6-Q16-windows-dll.exe del CD de instalación.

Configuración de Ghostscript 9.02 para Windows

Este software lo utiliza el servicio de HAPI para manipular archivos con formato pdf. Se debe editar el archivo \fdg\netappserver\gs9.02\bin\PDFA_def.ps y se debe modificar la línea "/ICCProfile (D:/HermesDSApp/NetAppServer/gs9.02/bin/default_cmyk.icc)" para poner la ruta correcta del archivo default_cmyk.icc.

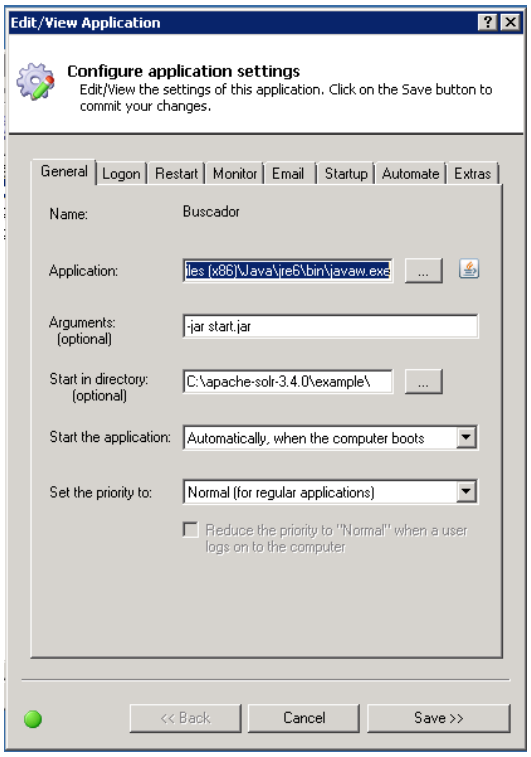
Instalación de tesseract 3.01 para Windows

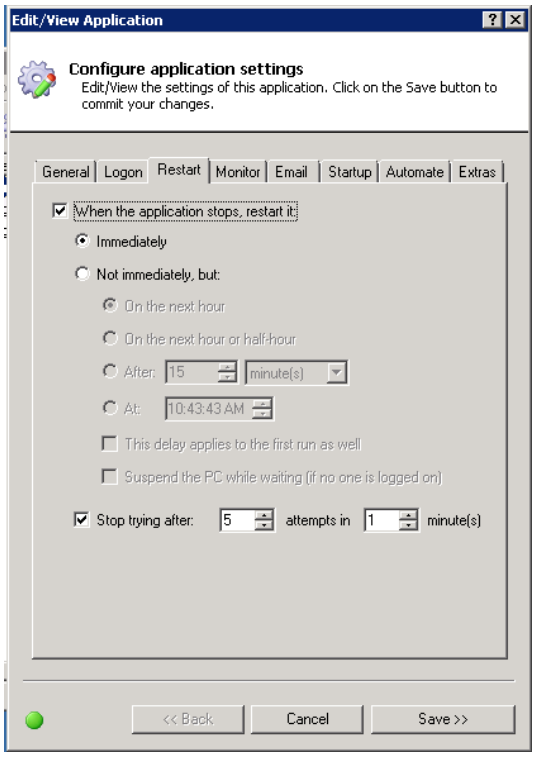
Este software lo utiliza el servicio de HAPI para realizar funciones de OCR. Se debe ejecutar el instalador \fdg\ tesseract-ocr-setup-3.01-1.exe que está en el CD de instalación

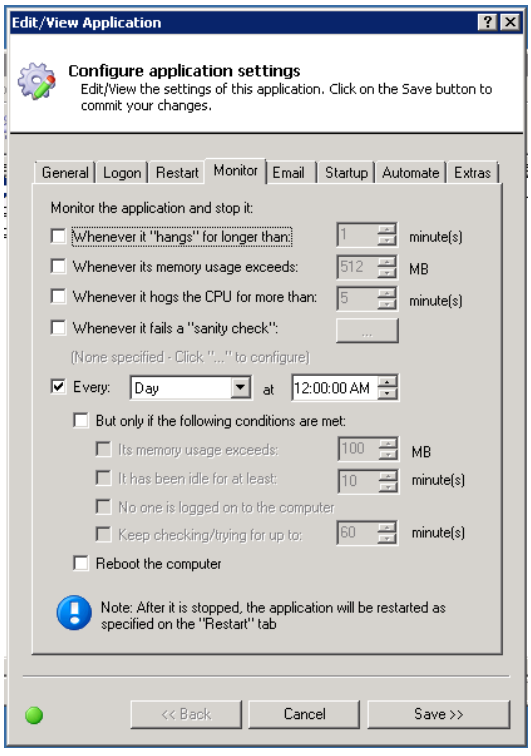
Instalación del solr

Este software es un motor de búsqueda que requiere el Java SE RT 1.6. Para instalar el software solr se deben seguir los siguientes pasos:

- Instalar Java SE RT 1.6. El instalador se encuentra en la ruta \fdg\ jre-6u32-windows-i586.exe del CD de instalación.
- Descomprimir el \fdg\archivo apache-solr-3.6.0.zip que está en el CD de instalación en c:\ del servidor de aplicaciones
- Configurar solr para que sea ejecutado por AlwaysUp con los siguientes parámetros:


Edit/View Application
Configure application settings
Edit/View the settings of this application. Click on the Save button to commit your changes.
General
Logon
Restart
Monitor
Email
Startup
Automate
Extras
Name:
Buscador
Application:
iles (x86)\Java\jre6\bin\javaw.exe
Arguments: (optional)
-jar start.jar
Start in directory: (optional)
C:\apache-solr-3.4.0\example\
Start the application:
Automatically, when the computer boots
Set the priority to:
Normal (for regular applications)
Reduce the priority to "Normal" when a user logs on to the computer
<< Back
Cancel
Save >>


Edit/View Application
Configure application settings
Edit/View the settings of this application. Click on the Save button to commit your changes.
General
Logon
Restart
Monitor
Email
Startup
Automate
Extras
When the application stops, restart it:
Immediately
Not immediately, but:
On the next hour
On the next hour or half-hour
After:
15
minute(s)
At:
10:43:43 AM
This delay applies to the first run as well
Suspend the PC while waiting (if no one is logged on)
Stop trying after:
5
attempts in
1
minute(s)
<< Back
Cancel
Save >>


Edit/View Application
Configure application settings
Edit/View the settings of this application. Click on the Save button to commit your changes.
General
Logon
Restart
Monitor
Email
Startup
Automate
Extras
Monitor the application and stop it:
Whenever it "hangs" for longer than:
1
minute(s)
Whenever its memory usage exceeds:
512
MB
Whenever it hogs the CPU for more than:
5
minute(s)
Whenever it fails a "sanity check":
(None specified - Click "..." to configure)
Every:
Day
at
12:00:00 AM
But only if the following conditions are met:
Its memory usage exceeds:
100
MB
It has been idle for at least:
10
minute(s)
No one is logged on to the computer
Keep checking/trying for up to:
60
minute(s)
Reboot the computer
Note: After it is stopped, the application will be restarted as specified on the "Restart" tab
<< Back
Cancel
Save >>

Ajustar parámetros de DSAppServer.exe.config

Se deben ajustar varias rutas del archivo de configuración del servicio de HAPI para reflejar la ruta completa del directorio de instalación del servicio de HAPI y se debe verificar que las rutas de los directorios y archivos existan:

- QueryPath. d:\fdg\queriesqueriesLog
- LogConfigFilename. d:\fdg\servidor\appLogging.config.xml
- CacheDir. d:\temp
- ExpFoFilename. d:\fdg\netappserver\temp-exp-pdf.fo
- ExportTempDir. d:\temp
- CRXSLFOPath. d:\fdg\netappserver\crxslfo.exe
- CRLDirectory. d:\temp
- DirAnexos. Directorio donde se guardan permanentemente los anexos subidos por medio de los formularios de HAPI . Por defecto es d:\fdg\anexos
- DirFormularios. Directorio donde se guardan los xml de los formularios definidos por el usuario. Por defecto es d:\fdg\formularios
- RutaTareasDinamicas. Ruta de archivo xml donde se guarda la definición de las tareas dinamicas introducidas por el usuario. Por defecto es D:\fdg\Formularios\TareasDinamicas.xml.
- RutaReportes. Ruta de archivo xml donde se guarda la definición de los reportes. Por defecto es D:\fdg\Formularios\Reportes.xml.
- WFDTOPath. Directorio donde se guardan los dtos de los flujos de trabajo. Por defecto es d:\fdg\dtos
- FlujosDeTrabajoPath. Ruta de archivo XML donde se guarda el índice de los flujos de trabajo definidos por el usuario. Por defecto es D:\fdg\cliente\FlujosDeTrabajo.xml.
- RutaDllsFlujos. Ruta donde se guardan los dlls de los flujos de trabajo compilados. Debe ser el mismo directorio “Plugins”. Por defecto es D:\fdg\servidor\Plugins.
- wfcPath. Ruta de la aplicación wfc.exe del framework de .NET. Por ejemplo: d:\fdg\Servidor\wfc.exe
- XslPath. D:\fdg\Cliente
- RutaTipoSolicitud. D:\fdg\Cliente
- RutaDocumentosFirmados. D:\fdg\anexos.
- ValidadoresCamposTextoPath. Ruta de archivo con parámetros para campos de texto especiales. Por defecto es d:\fdg\Cliente\Validadores.xml
- HafFormToXslFoPath. Ruta del archivo para convertir un formulario a formato pdf. Por defecto es D:\fdg\Clientes\HafFormToXslFo.xsl
- HtmlEntitiesPath. Ruta del archivo con mapa de entidades html y su respectiva letra o simbolo. Por defecto es D:\fdg\Cliente\HtmlEntities.txt

- ConvertToPDFA. Parámetro que controla si los formularios convertidos a pdf deben usar el formato pdfa o no. Los posibles valores son “si” o “no”. Nótese que la conversión a pdfa puede ser muy lenta.

El valor de los siguientes parámetros corresponde al directorio donde se guardan los textos de las notificaciones definidas por los usuarios.

- RutaTipoNotificacion. Esta es la ruta donde se guardan los textos de las notificaciones. Por defecto es D:\fdg\Cliente\.
- RutaNotificacionDefault. Debe tener el mismo valor que RutaTipoNotificacion.

El valor de los siguientes parámetros corresponde a la ruta de diferentes plantillas de notificaciones utilizadas durante el proceso de activación de una cuenta por email.

- NotiRequiereActivacionPlantilla. Por defecto es D:\fdg\Cliente\NotiRequiereActivacion.html.
- NotiRecordatorioActivacionPlantilla. por defecto es D:\fdg\Cliente\NotiRecordatorioActivacion.html
- NotiActivacionCuentaPlantilla. Por defecto es D:\fdg\Cliente\NotiActivacionCuenta.html.
- NotiCancelacionCuentaPlantilla. Por defecto es D:\fdg\Cliente\NotiCancelacionCuenta.html
- NotificacionGeneralPlantilla. Por defecto es D:\fdg\Cliente\PlantillaNotificacion.html.
- NotiCambioClavePlantilla. Por defecto es D:\fdg\Cliente\NotiCambioClave.html

Debe verificarse que el usuario “Network Service” tiene permisos full-control sobre todos los anteriores archivos y directorios anteriores.

Los siguientes parámetros son URLs de páginas de HAPI que debe actualizarse para reflejar el nombre del dominio del servidor web del cliente y el directorio virtual de la aplicación web.

- URLActivacion. URL de la página AutenticacionEmail.aspx
- URLProcesamientoTarea. URL de la página ProcesamientoTarea.aspx
- URLCompletarFormulario. URL de la página CompletarFormulario.aspx.
- URLSeguimientoSolicitud. URL de la página SeguimientoSolicitud.aspx

Los siguientes parámetros corresponden a intervalos de tiempo en minutos usados durante el proceso de activación de una cuenta por medio de email

- MinutosExpiraActivacion. Intervalo para cancelar la activación de una cuenta nueva si el usuario lo la activa. Por defecto es 240, es decir 4 horas.
- MinutosParaRecordatorioActivacion. Intervalo para envío de recordatorio de activación de cuenta. Por defecto es 60, es decir una hora.

El valor de los siguientes parámetros debe cambiarse por un email del cliente que se utilizará en el “from” de varias notificaciones que envía el sistema. Típicamente este email es el mismo pero podrían ponerse diferentes emails.

- NotiRequiereActivacionEmailEnvia
- NotiRecordatorioActivacionEmailEnvia
- NotiActivacionCuentaEmailEnvia
- NotiCancelacionCuentaEmailEnvia
- NotificacionGeneralEmailEnvia
- NotiCambioClaveEmailEnvia

El valor de los siguientes parámetros refleja el asunto que se pondrá en diferentes notificaciones generadas durante el proceso de activación de una cuenta por email.

- NotiRequiereActivacionAsunto
- NotiRecordatorioActivacionAsunto
- NotiActivacionCuentaAsunto
- NotiCancelacionCuentaAsunto
- NotificacionGeneralAsunto
- NotiCambioClaveAsunto

Los siguientes parámetros se utilizan para el servidor de correo electrónico y deben ajustarse para que el sistema pueda enviar emails.

- SmtpServer. Dirección IP del servidor de SMTP
- SmtpPort. Puerto del servicio de SMTP, tipicamente el 25
- SmtpUser. Nombre del usuario utilizado para enviar email
- SmtpUserPassword. Clave encriptada del usuario.
  Para generar la clave encriptada debe usarse la aplicación SCConfig.exe con la opción "Usar llave interna marcada".

Rutas y parámetros para aplicaciones externas que deben ajustarse dependiendo de las rutas donde hayan quedado instalados los programas auxiliares.

- RutaImageConverter. Ruta de la aplicación convert.exe de ImageMagick
- GsWin32cPath. Ruta de la aplicación gswin32c.exe. Por defecto es D:\fdg\NetAppServer\gs9.02\bin\gswin32c.exe
- GsWin32cArguments. Ruta completa del directorio donde están los ejecutables de Ghostscript. Se debe ajustar el valor para el parámetro "-ID" que por defecto es "D:/fdg/NetAppServer/gs9.02/bin". Nótese que se debe usar "/" en lugar de "\".
- GsWin32cResourcePath. Ruta completa del directorio donde están los ejecutables de Ghostscript. Por defecto es "D:/fdg/NetAppServer/gs9.02/bin". Nótese que se debe usar "/" en lugar de "\".
- TesseractFilepath. Ruta completa al ejecutable principal de Tesseract: tesseract.exe.
- TesseractLogFilepath. Ruta completa al log de tesseract: tesseract.log.

Otros parámetros:

- DigSigVersion. Versión del componente de firma digital.
- AgenteElectronicoCAId. 1006 si se utiliza un certificado de SINPE o 1005 si se utiliza un certificado de Hermes para la firma con agente electrónico

Personalizar diseño grafico de las notificaciones

Los siguientes archivos html que se encuentran en el subdirectorio “cliente” contienen las plantillas de las notificaciones que envía el sistema:

- NotiActivacionCuenta.html
- NotiCambioClave.html
- NotiCancelacionCuenta.html
- NotiFirmar.html
- NotiFirmasCompletadas.html
- NotiRecordatorioActivacion.html
- NotiRequiereActivacion.html
- PlantillaNotificacion.html
- RevisarFormulario.html
- TareaAsignada.html

Por lo menos debe actualizarse el URL de las imágenes para que apunten al sitio web del cliente.

Ajustar rutas de HafFormToXslFo.xsl

Se deben ajustar las rutas a las imagenes dentro del archivo \fdg\cliente\ HafFormToXslFo.xsl para reflejar la ruta de instalación. Las rutas a las imágenes por defecto están en “D:\HermesDSApp\HermesDSSite\Images”.

Conexión a base de datos

Se deben editar todas las hileras de conexión del archivo d:\fdg\servidor\DSAppServer.exe.config para poner la dirección del servidor de base de datos y los parámetros de autenticación:

- “Data Source=” o “server=” : dirección ip del servidor de base de datos
- “Initial Catalog=” o “database=” : Nombre de la base de datos, por defecto HermesDSApp
- “User ID=” o “uid=” : nombre del usuario de base de datos
- “Password=” or “pwd=” : clave del usuario de base de datos

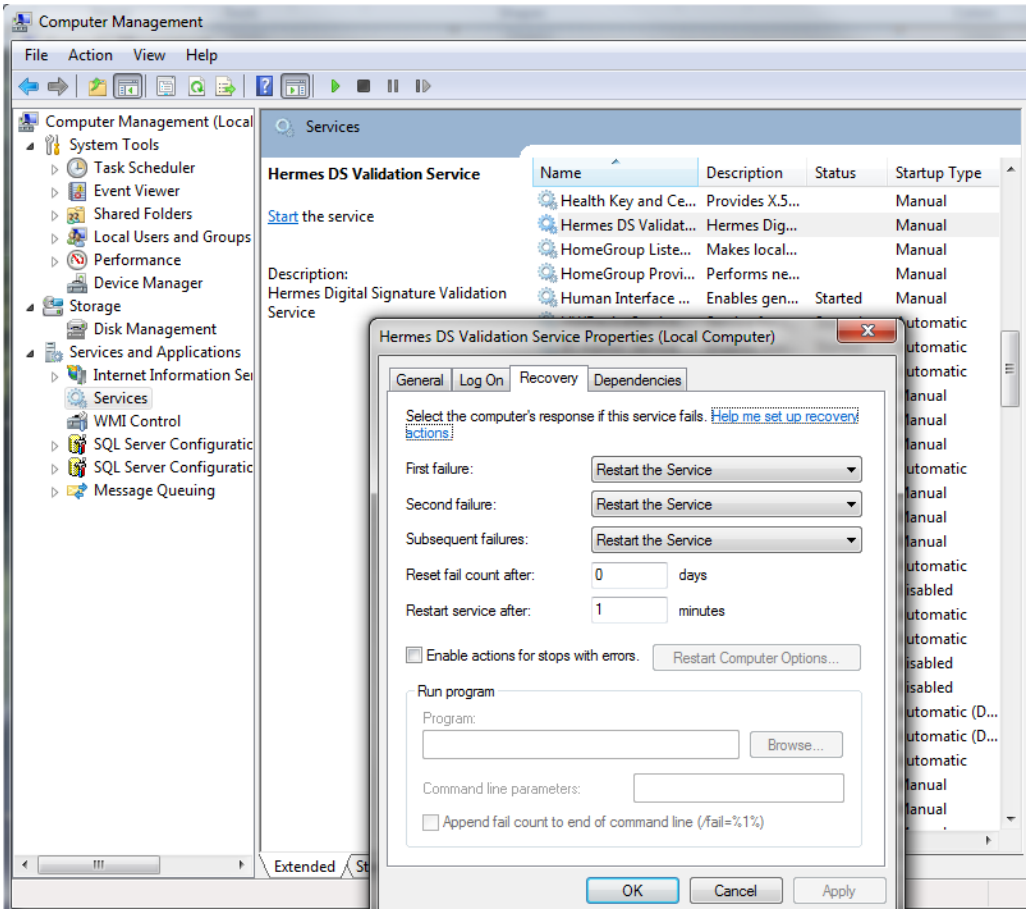
Asignar puerto usado por servidor de aplicaciones

- Modificar la configuración de remoting del archivo d:\fdg\servidor\DSAppServer.exe.config para poner el número de puerto a utilizar por el servicio de validación de firma digital:

```
<system.runtime.remoting>
  <application name="DSAppService">
    <service>
      <wellknown mode="SingleCall" type="RequestHandl..." />
    </service>
    <channels>
      <channel ref="tcp" port="43210" displayName="..." />
...
```

Instalar y configurar el servicio de HAPI

- Ejecutar el command prompt (cmd)
- Ejecutar el comando
  set PATH=%PATH%;C:\Windows\Microsoft.NET\Framework\v4.0.30319
- Utilizar el comando “cd” para Ir al directorio d:\fdg\servidor
- Ejecutar el comando “installutil /i DSAppServer.exe”. En la lista de servicios de Windows debe aparecer el servicio “Hermes DS Validation Service”.
- En la pestaña de “Log on” asegurarse que el servicio levante con el usuario “Network Service”
- En la pestaña de “General” asegurarse que el modo de inicio del servicio es automatico
- En la pestaña de “Recovery” asegurarse que todas las opciones de recovery del servicio están en “Restart the service”

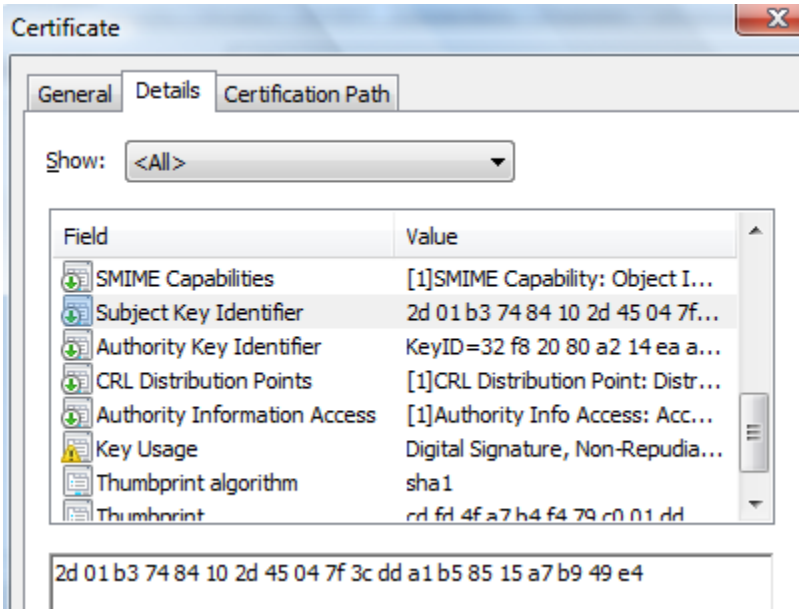


Ajustar parámetros del servidor de aplicaciones (DSAppServer.exe.config)

- GMTOffset. Cantidad de horas de desplazamiento respecto a GMT. Por defecto es -6.
- OCSPRefreshInterval. Cantidad de segundos a esperar para volver a consultar el servicio OCSP para un certificado específico. El valor por defecto es 600 segundos.
- CRLRefreshInterval. Cantidad de segundos a esperar para descargar la lista de revocación de certificados. El valor por defecto es 3600 segundos.
- TsaUrl. URL del servicio de estampado de tiempo, por ejemplo:
  Este parámetro debe actualizarse con la dirección del servicio de estampado de tiempo del SINPE una vez que dicho servicio esté en producción

Parámetros del servicio de firma digital de agente electrónico

- PdfCertSubjKeyId. Este parámetro identifica el certificado de agente electrónico de firma digital que se utilizará para firmar digitalmente los comprobantes. Este parámetro corresponde al campo “Subject Key Identifier” del certificado, removiendo todos los espacios, por ejemplo: 2D01B37484102D45047F3CDDA1B58515A7B949E4



En la sección Instalación de un certificado de agente electrónico se explica cómo instalar y configurar un certificado de manera que sea accesible para el servicio de validación de firma digital

El certificado \fdg\HAPI.pfx se puede utilizar como certificado de agente electrónico para efectos de pruebas.

- XslPath. Este parámetro es la ruta donde se guardarán las plantillas xsl utilizadas para generar los comprobantes. Por defecto es d:\fdg\cliente.

Configuración del proxy

En esta sección se explican los pasos para configurar el servidor de aplicaciones para que acceda mediante un proxy los servicios de OCSP, CRLs y Estampado de tiempo. Estos pasos son necesarios solo si se requiere utilizar un proxy para salir a Internet desde el servidor de aplicaciones.

1. Se debe incluir la siguiente sección en el elemento <configuration> del DSAppServer.exe.config

```
<configuration>

...

        <system.net>

                <defaultProxy>

                        <proxy usesystemdefault="False"
                proxyaddress="http://192.168.1.7:8080"
                 bypassonlocal="False"/>

                </defaultProxy>

        </system.net>

...

</configuration>
```

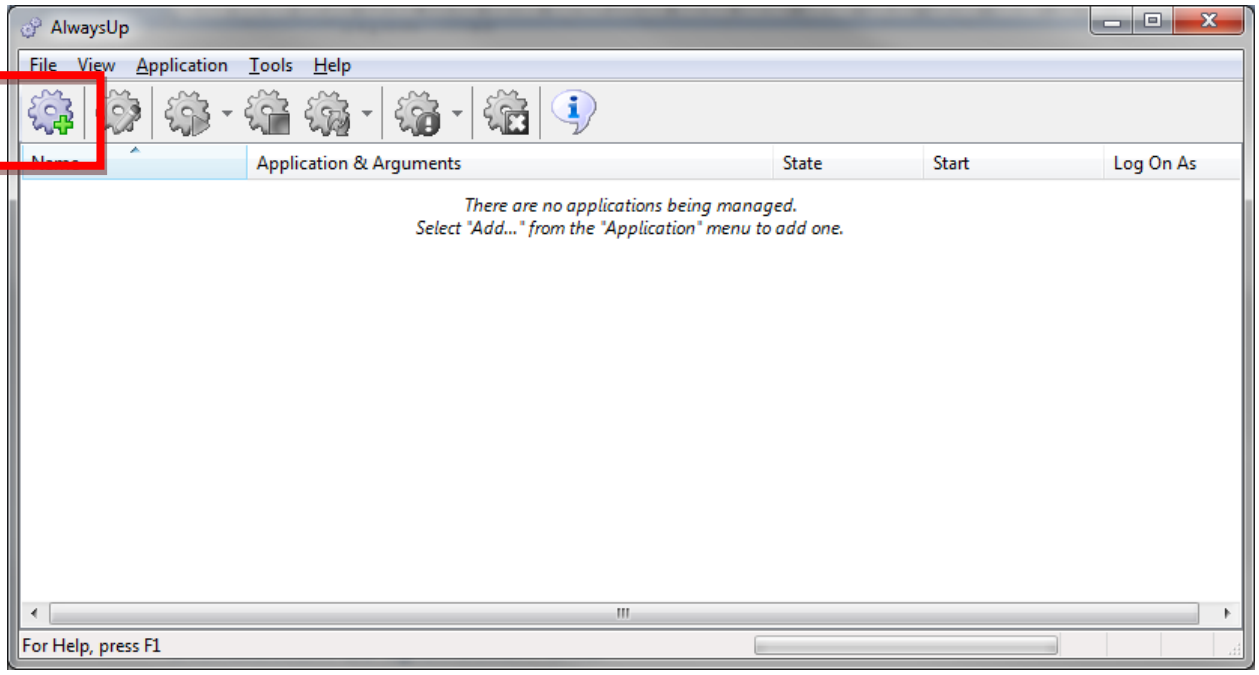
2. Se debe modificar el valor del parametro TsaUrl para incluirle la dirección y puerto del proxy al final de la dirección de la siguiente forma @192.168.1.7,8080:.

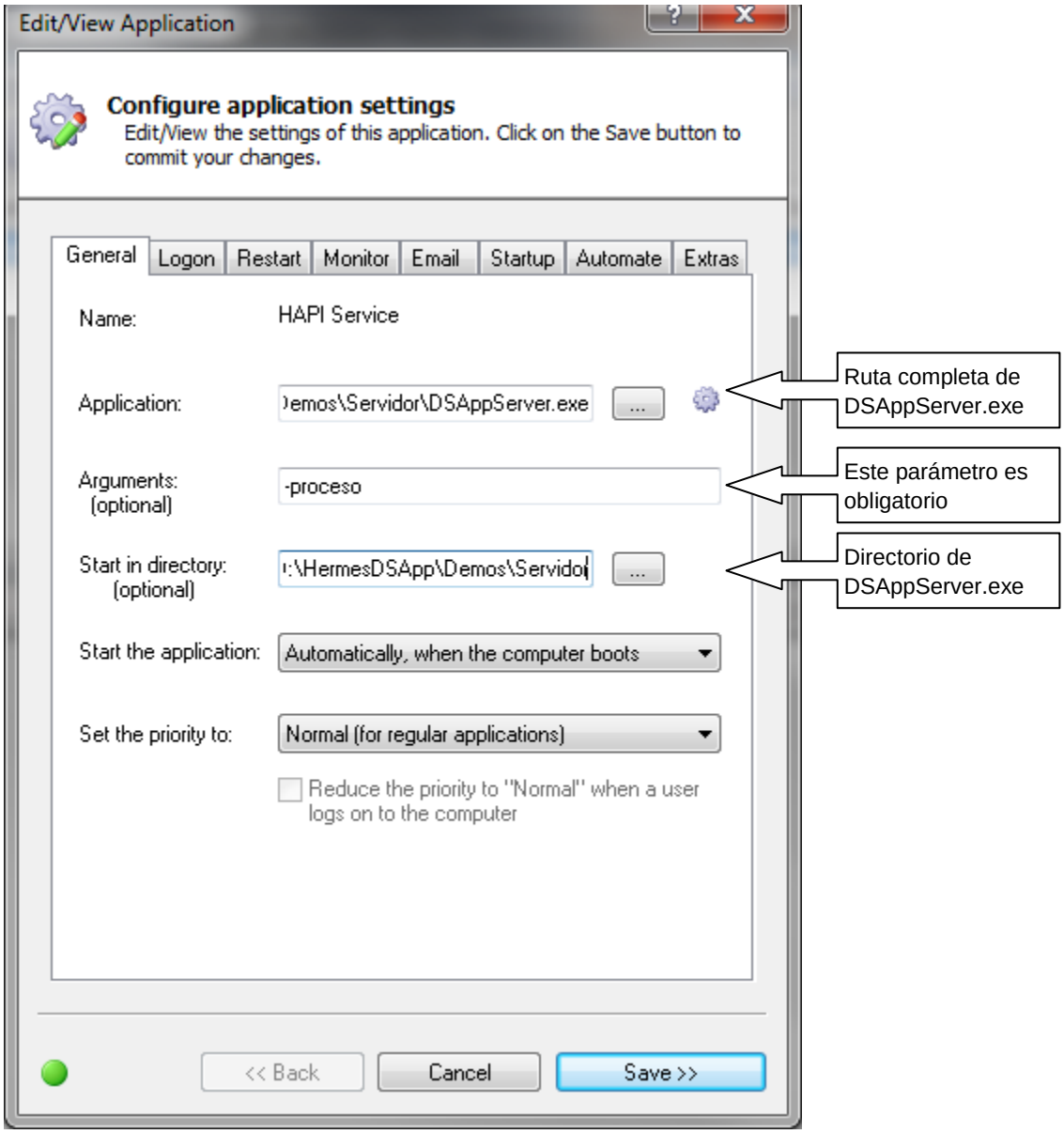
   <add key="TsaUrl?" value="http://www.ca-soft.com/request.aspx**@192.168.1.7,8080**"/>

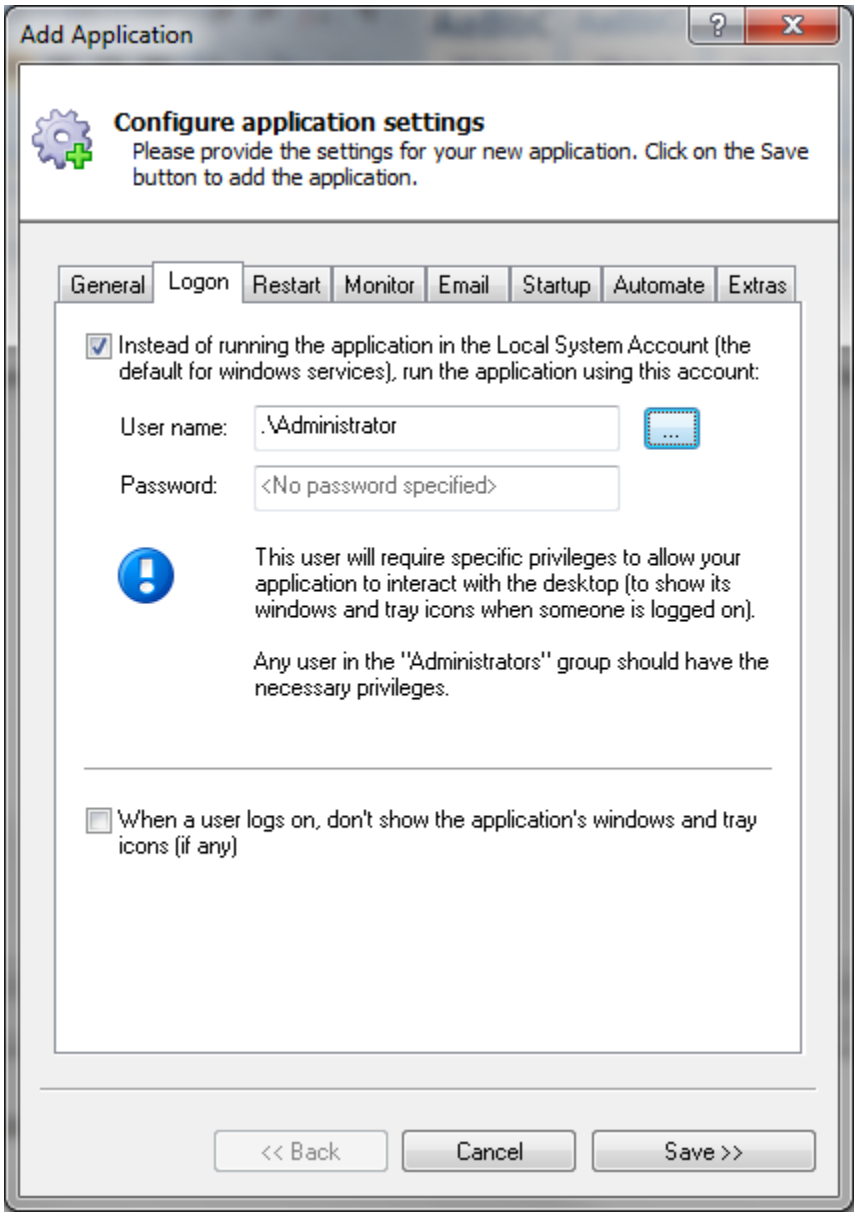
En los anteriores ejemplos la dirección del proxy es 192.168.1.7 y el puerto del proxy es 8080

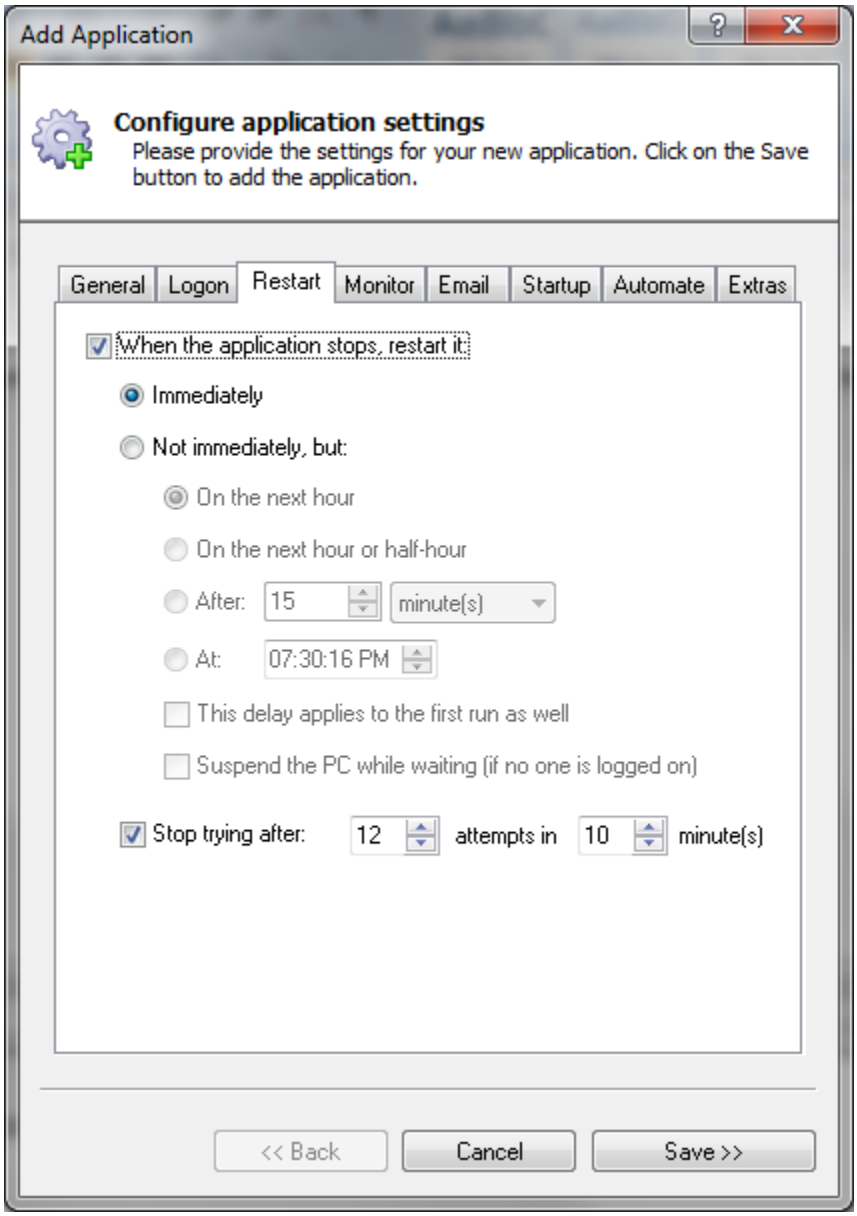
Instalación del servidor de aplicaciones con AlwaysUp

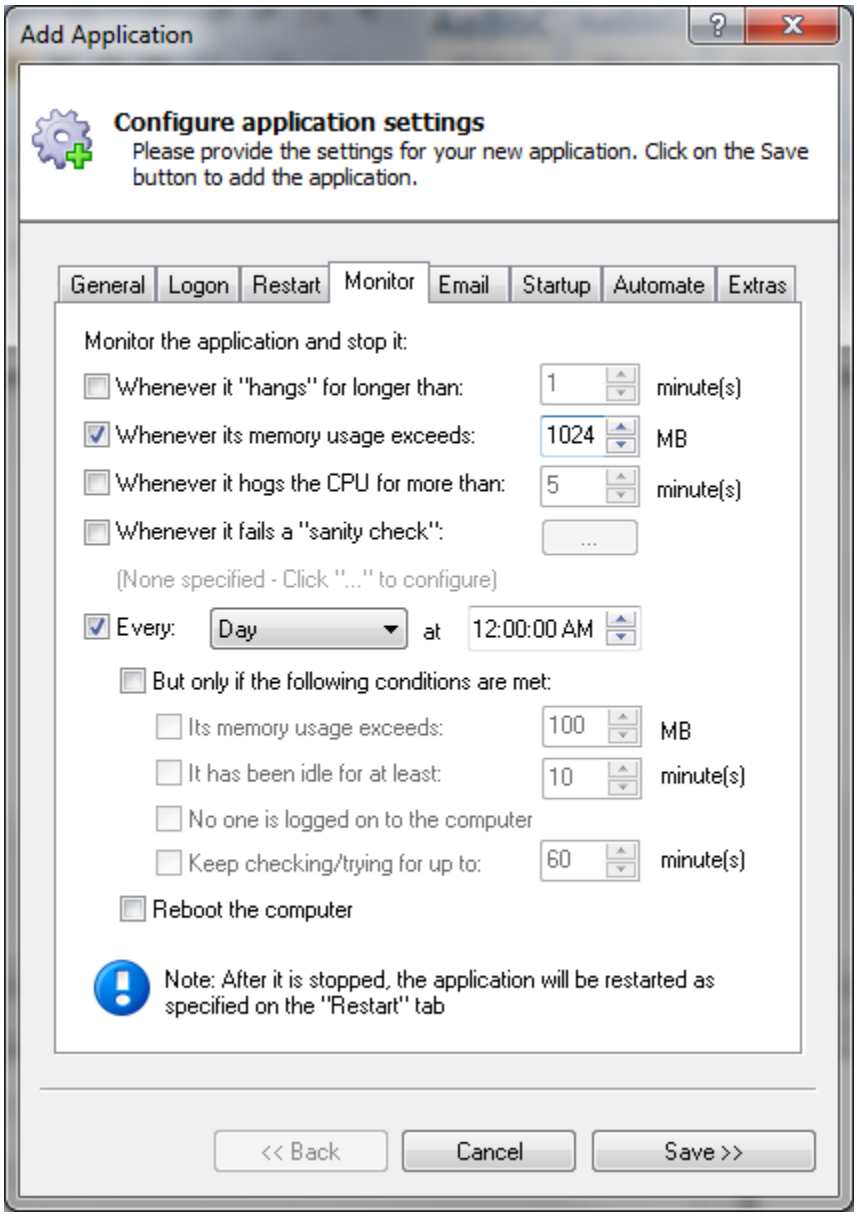
Al correr el servidor de aplicaciones con AlwaysUp se garantiza la disponibilidad permanente del servicio. A continuación se muestran las pantallas de configuración:

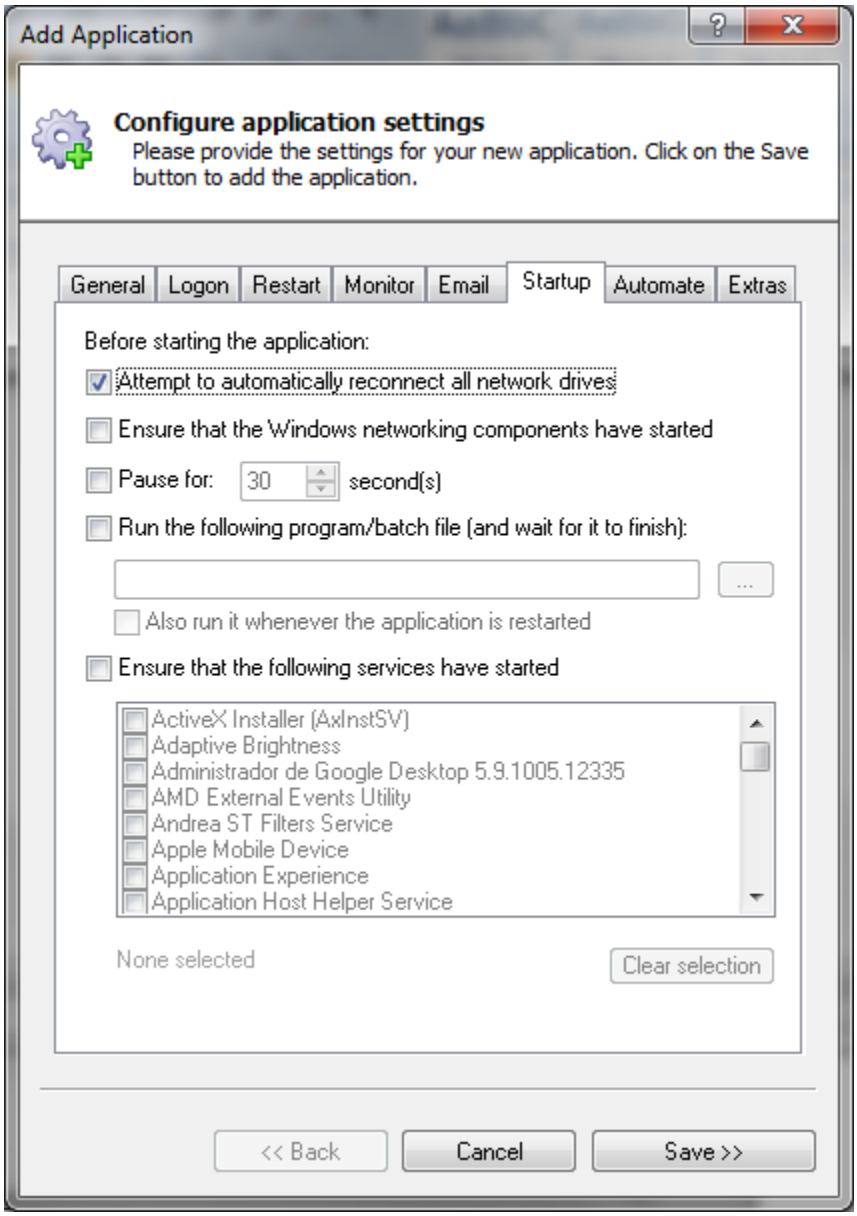

AlwaysUp
File View Application Tools Help
Application & Arguments
State
Start
Log On As
There are no applications being managed.
Select "Add..." from the "Application" menu to add one.
For Help, press F1

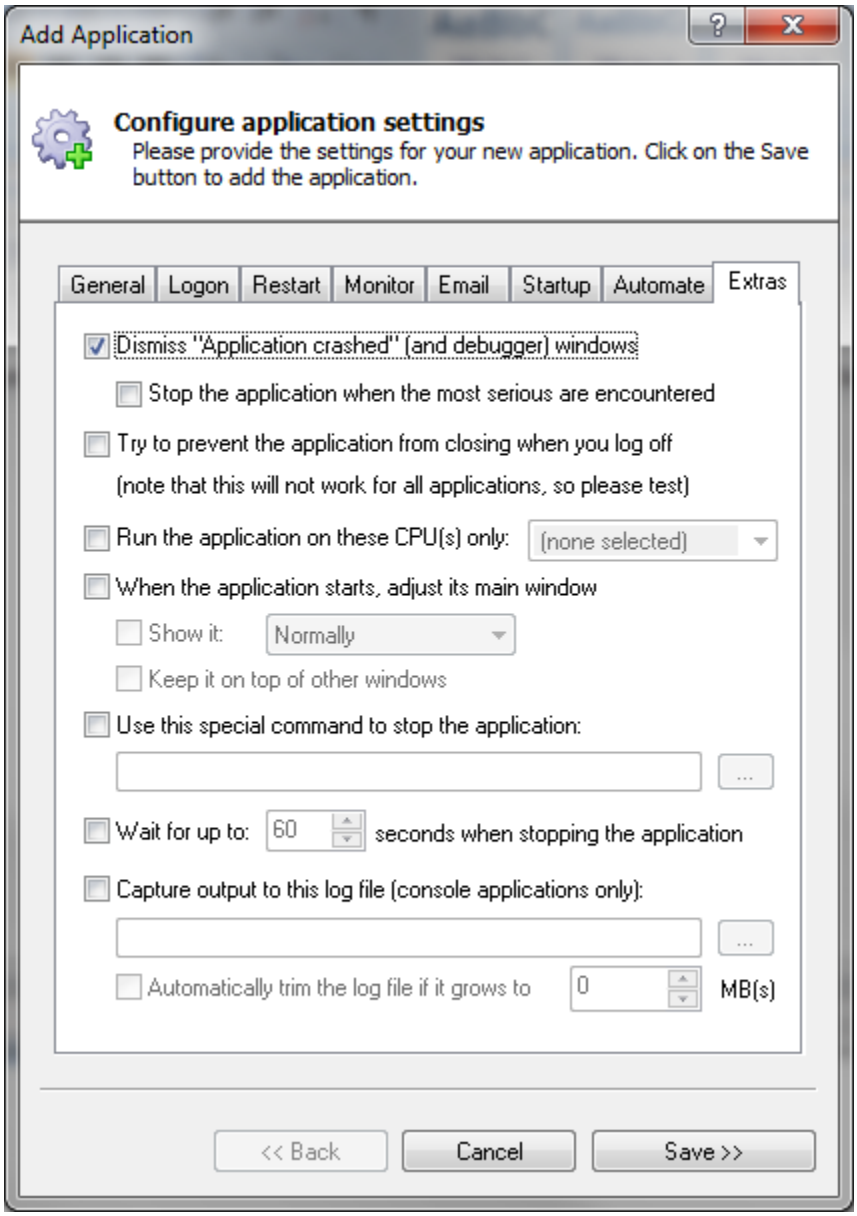

Edit/View Application
Configure application settings
Edit/View the settings of this application. Click on the Save button to commit your changes.
General
Logon
Restart
Monitor
Email
Startup
Automate
Extras
Name:
HAPI Service
Application:
)emos\Servidor\DSAppServer.exe
...
Arguments: (optional)
-proceso
Start in directory: (optional)
I:\HermesDSApp\Demos\Servidor
...
Start the application:
Automatically, when the computer boots
Set the priority to:
Normal (for regular applications)
Reduce the priority to "Normal" when a user logs on to the computer
<< Back
Cancel
Save >>
Ruta completa de DSAppServer.exe
Este parámetro es obligatorio
Directorio de DSAppServer.exe


Add Application
Configure application settings
Please provide the settings for your new application. Click on the Save button to add the application.
General
Logon
Restart
Monitor
Email
Startup
Automate
Extras
Instead of running the application in the Local System Account (the default for windows services), run the application using this account:
User name:
.\Administrator
...
Password:
<No password specified>
This user will require specific privileges to allow your application to interact with the desktop (to show its windows and tray icons when someone is logged on).
Any user in the "Administrators" group should have the necessary privileges.
When a user logs on, don't show the application's windows and tray icons (if any)
<< Back
Cancel
Save >>


Add Application
Configure application settings
Please provide the settings for your new application. Click on the Save button to add the application.
General
Logon
Restart
Monitor
Email
Startup
Automate
Extras
When the application stops, restart it:
Immediately
Not immediately, but:
On the next hour
On the next hour or half-hour
After:
15
minute(s)
At:
07:30:16 PM
This delay applies to the first run as well
Suspend the PC while waiting (if no one is logged on)
Stop trying after:
12
attempts in
10
minute(s)
<< Back
Cancel
Save >>


Add Application
Configure application settings
Please provide the settings for your new application. Click on the Save button to add the application.
General
Logon
Restart
Monitor
Email
Startup
Automate
Extras
Monitor the application and stop it:
Whenever it "hangs" for longer than:
1
minute(s)
Whenever its memory usage exceeds:
1024
MB
Whenever it hogs the CPU for more than:
5
minute(s)
Whenever it fails a "sanity check":
...
(None specified - Click "..." to configure)
Every:
Day
at
12:00:00 AM
But only if the following conditions are met:
Its memory usage exceeds:
100
MB
It has been idle for at least:
10
minute(s)
No one is logged on to the computer
Keep checking/trying for up to:
60
minute(s)
Reboot the computer
Note: After it is stopped, the application will be restarted as specified on the "Restart" tab
<< Back
Cancel
Save >>


Add Application
Configure application settings
Please provide the settings for your new application. Click on the Save button to add the application.
General
Logon
Restart
Monitor
Email
Startup
Automate
Extras
Before starting the application:
Attempt to automatically reconnect all network drives
Ensure that the Windows networking components have started
Pause for:
30
second(s)
Run the following program/batch file (and wait for it to finish):
...
Also run it whenever the application is restarted
Ensure that the following services have started
ActiveX Installer (AxInstSV)
Adaptive Brightness
Administrador de Google Desktop 5.9.1005.12335
AMD External Events Utility
Andrea ST Filters Service
Apple Mobile Device
Application Experience
Application Host Helper Service
None selected
Clear selection
<< Back
Cancel
Save >>



Instalación de aplicación web

Los siguientes son los pasos para instalar la aplicación web en el servidor web:

Copiar archivos de la aplicación web

- Copiar en el directorio raíz del sitio web, generalmente c:\inetpub\wwwroot, las subcarpetas js e images que se encuentran en el directorio \fdg del CD de instalación
- Copiar en el directorio d:\fdg del servidor web las carpetas \fdg\fdg, \fdg\ NetAppServer, \fdg\formularios y \fdg\cliente que se encuentran en el CD de instalación
- Otorgar permiso de lectura y escritura al usuario con el que corre el IIS, típicamente “Network Service” en el directorio d:\fdg\fdg\graphimages, en el caso de IIS 7.5 se debe dar permisos al grupo IIS_IUSRS.

Registrar aplicación web en el IIS

- En el IIS crear un pool de aplicaciones llamado “fdg”
- Desde el administrador del IIS crear una aplicación llamado HermesDSApp que apunte hacia el directorio d:\fdg\fdg y asignarle el pool de aplicaciones “fdg”
- Configurar el directorio virtual HermesDSApp para correr con el framework 4 de .Net
- Crear directorio virtual /images que apunte a d:\fdg\fdg\images
- Crear directorio virtual /hapi que apunte a d:\fdg\fdg
-

Definir documento por defecto

- Desde la consola de administración del IIS definir el documento por defecto de HermesDSApp, a saber, el archivo default.aspx

Copiar dlls de ReportViewer

- Copiar en el directorio d:\fdg\fdg\bin los archivos que se encuentra en el directorio \fdg\ReportViewer del CD de instalación: Microsoft.ReportViewer.WebForms.dll, Microsoft.ReportViewer.Common.dll, Microsoft.ReportViewer.ProcessingObjectModel.dll

Instalar el runtime de Visual J

- Ejecutar vjredist.exe. Estas librerías son necesarias para las funciones de exportación.

Registrar el componente servidor de firma digital.

- regsvr32 d:\fdg\digsigservercom.dll. Este dll se encuentra en la carpeta “d:\fdg”.

Actualizar dirección del servidor de aplicaciones

- Poner la dirección del servidor de aplicaciones o del NLB en la sección RemoteServers del archivo d:\fdg\fdg\web.config

```
<RemoteServers>
   <Server Active="true" Url="tcp://127.0.0.1:43210/RemoteRequestServer"/>
</RemoteServers>
```

Ajustar rutas de web.config

Se deben revisar varias rutas en el archivo de configuración de la aplicación web para verificar que las mismas existan. Si se instalo en el directorio d:\fdg entonces las rutas deberían estar bien, pero si se instaló en otro directorio entonces las rutas deben modificarse para reflejar la ruta de instalación elegida:

- ExpFoFilename. d:\fdg\netappserver\temp-exp-pdf.fo
- RutaCRDoc2PdfConverter. D:\fdg\NetAppServer\CRDoc2PdfConverter.exe
- ExportTempDir. d:\temp
- CRXSLFOPath. d:\fdg\netappserver\crxslfo.exe
- URLDigSigSetup. /hermesdsapp/digsigsetup.exe#VERSION=1,27,0,0
- Sitio. Nombre del sitio web.
- mostrarMenuRegistro. “true” para permitir la inscripción de usuarios y “false” para no permitirla
- RutaSnippets. Directorio donde se guardan varias plantillas de actividades usadas por el editor de flujos de trabajo. Por defecto es D:\fdg\Cliente
- RutaFormularioVacio . Ruta de archivo con definición de formulario por defecto usado cuando se crea un nuevo formulario D:\fdg\Formularios\FormularioVacio.xml
- RutaPdfs. d:\Temp
- RutaTareaDinamicaNueva. D:\fdg\Formularios\TareaDinamicaNueva.xml
- RutaTipoNotificacion. Ruta al directorio donde están los html de las notificaciones. Por defecto es d:\fdg\cliente.

- HtmlEntitiesPath. Ruta del archivo con mapa de entidades html y su respectiva letra o simbolo. Por defecto es D:\fdg\Cliente\HtmlEntities.txt
- RutaLogoSitio. Url relativo a imagen con logo primario del cliente
- RutaLogoDerechaSitio. Url relativo a imagen con logo secundario del cliente
- MemcachedServers. Dirección ip y puerto de la instancia de memcached que fue instalada en el servidor de aplicaciones. Por defecto es 127.0.0.1:11211.

Poner ruta de webservice de validación

- Es necesario modificar el web.config para especificar la dirección del webservice de validación:

```
<applicationSettings>
  <HermesDSSite.Properties.Settings>
    <setting name="HermesDSSite_localhost_ServicioFirmaDigital" SerializeAs="String">
        <value>....</value>
    </setting>
  </HermesDSSite.Properties.Settings>
</applicationSettings>
```

Instalar el componente cliente de firma digital.

- Ejecutar regsvr32 d:\fdg\NetAppServer\digsigcom.dll
- Cargar los archivos del registro de Windows: digsigcom-localmachine.reg y IDigitalSignatureComponente-localmachine.reg. *Estos archivos se encuentran en la carpeta fdg del servidor de aplicaciones.*

Instalación de Microsft Enterprise Library 5

Este software contiene varios componentes de Microsoft que son utilizados por HAPI. Para instalar este software se debe ejecutar el instalador que está en la ruta \fdg\Enterprise Library 5.0.msi del CD de instalación.

Configurar sección mailSettings

- Se debe configurar el parametro from, host, password y userName para garantizar que se pueda enviar el correo de activación de cuentas, por ejemplo:

```
<system.net>
```

```
    <mailSettings>
      <smtp from="b.simon@hermes-soft.com">
        <network host="196.40.71.23" password="123456"
userName="b.simon"/>
      </smtp>
    </mailSettings>
  </system.net>
```

Copiar dlls de exportación a pdf en System32

- Copiar los dlls \fdg\netappserver\WRW3.DLL y \fdg\netappserver\WRS3.DLL en System32 en el caso de Windows 32 Bits o en SysWOW64 en el caso de windows 64 bits.

Creación de usuario administrador

Ir a la dirección http://localhost/HermesDSApp y registrar al usuario administrador. Una vez que se registro exitosamente el usuario ejecutar el siguiente script:

```
insert into aut.RolesDeUsuario(fromUsuarioId,toRolid)
select 1, RolId
from aut.Rol
```

El anterior script le asigna al usuario todos los roles posibles, lo cual lo convierte en el usuario administrador.

Creación de horario “Normal”

Se debe crear el horario “estándar” que es utilizado por el sistema para cálculos de duraciones. Dicho horario debe ingresarse con el nombre “Normal” y debe reflejar el horario normal de trabajo del cliente. El horario debe abarcar por lo menos 10 años a futuro.

Configuración de la autoridad certificadora

Tanto en el servidor de aplicaciones como en el servidor web se tiene que configurar la autoridad certificadora a utilizar, la cual debe ser la misma para ambos servidores. Los pasos para configurar la autoridad certificadora son:

Poner el número de autoridad certificador a utilizar en los parámetros IntCAId y ExtCAId en el archivo de configuración de la aplicación web y en el archivo de configuración del servicio de validación de firma digital:

```
<add key="IntCAId" value="1006"/>
<add key="ExtCAId" value="1006"/>
```

Los posibles números son 1004 para la autoridad certificadora de prueba del SINPE y 1006 para la autoridad certificadora de producción del SINPE.

Poner el nombre de la autoridad certificador a utilizar en el parámetro IssuerDN en el archivo de configuración de la aplicación web y en el archivo de configuración del servicio de validación de firma digital:

```
<add key="IssuerDN" value="CN=PRUEBAS - CA EMISORA PERSONA FISICA,OU=SINPE,O=BCCR,C=CR,serialNumber=CPJ-9-999-999999"/>
```

Los posibles valores son “CN=PRUEBAS - CA EMISORA PERSONA FISICA,OU=SINPE,O=BCCR,C=CR,serialNumber=CPJ-9-999-999999” y “CN=CA SINPE - PERSONA FISICA,OU=DIVISION DE SERVICIOS FINANCIEROS,O=BANCO CENTRAL DE COSTA RICA,C=CR,serialNumber=4-000-004017” para la autoridad certificadora de prueba del SINPE y la autoridad certificadora de producción del SINPE respectivamente.

Registro de extensión .fhp en IIS 7

La extensión .fhp es utilizada para activar el firmador de HAPI y se debe registrar en el IIS como requisito para poder integrar las aplicaciones web con el Firmador de HAPI.

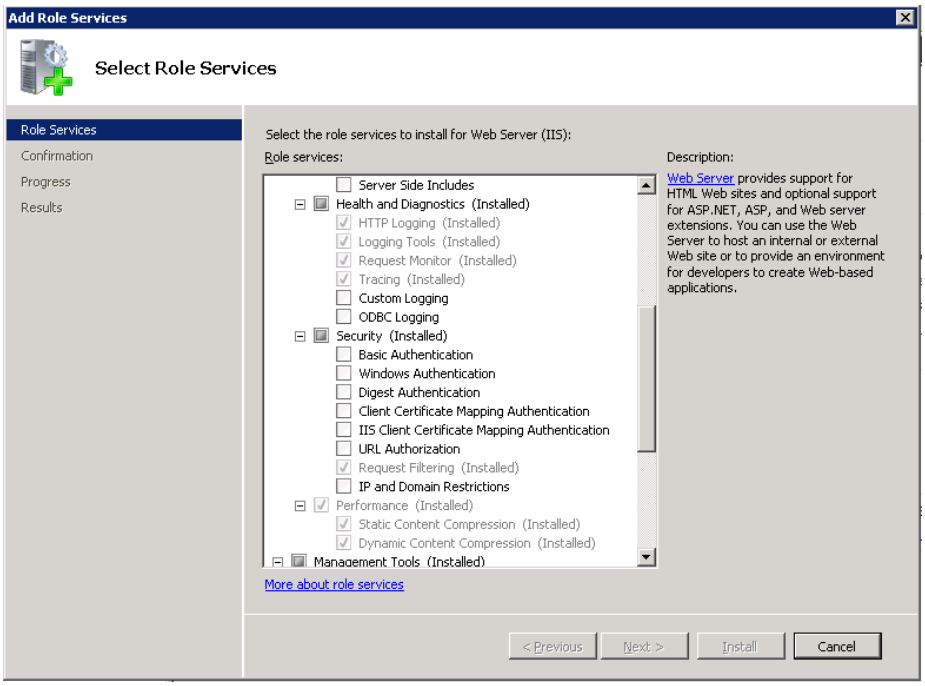
Para registrar la extensión .fhp en el IIS 7 se deben seguir los siguientes pasos:

1. Ir a “Administrative Tools” y hacer doble clic en el icono “Internet Information Services (IIS) Manager”
2. Seleccionar el directorio virtual HermesDSApp y hacer doble clic en el icono “MIME Types”
3. Hacer clic en el enlace “Add...” que aparece en el panel “Actions” en el extremo derecho
4. En el campo “File name extension” del dialogo “Add MIME Type” introducir “.fhp” y en el campo “MIME Type” introducir “application/x-hermes-hapi-firmador”
5. Por último hacer clic en el botón OK para registrar la extensión

Activar compresión en el IIS para HermesDSApp

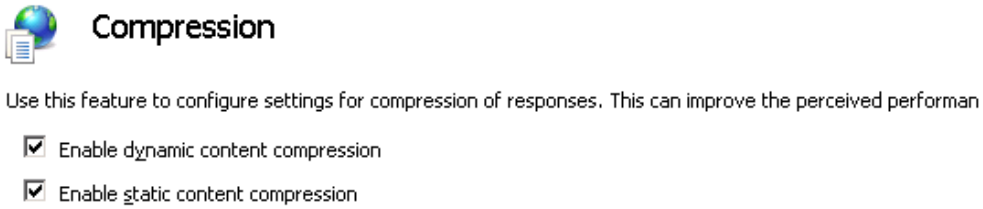
Este procedimiento explica como activar y configurar la compresión de datos para la aplicación HermesDSApp.

- Primero hay que asegurarse que el módulo de compresión está instalado.
    i. Abrir el server manager (Control Panel/Administrative Tools)
    ii. Expandir el nodo “Roles” y seleccionar el nodo “Web Server (IIS)”
    iii. Hacer clic en el enlace “Add Roles Services”. Se muestra un dialogo con el titulo “Select Role Services”
    iv. Marcar los checkbox “Static content compression” y “Dynamic content compression” que están en el nodo “Performance”
    v. Hacer clic en el botón “Install”



- Segundo se debe activar la compresión para la aplicacion web “HermesDSApp”
    i. Abrir el “Internet Information Services Manager” (Control Panel/Administrative tools)
    ii. Seleccionar el directorio virtual “HermesDSApp” (Sites/Default Web site/HermesDSApp)
    iii. Hacer doble clic en el icono “Compression” que aparece en la sección “IIS” del cuadro “Features view”

iv. Marcar los checkbox “Enable dynamic content compression” y “Enable static content compression”



v. Hacer clic en el enlace “Apply” que aperece en el panel derecho “Actions”

- Tercero se debe activar la compresión para contenido de tipo json
  i. Editar el archivo “applicationHost.config” (C:\Windows\System32\inetsrv\config)
  ii. Incluir los siguientes elementos en la sección “staticTypes” del elemento httpCompression

```
<add mimeType="application/json" enabled="true" />
<add mimeType="application/json; charset=utf-8" enabled="true" />
```

  iii. Incluir los siguientes elementos en la sección “dynamicTypes” del elemento httpCompression

```
<add mimeType="application/json" enabled="true" />
<add mimeType="application/json; charset=utf-8" enabled="true" />
```

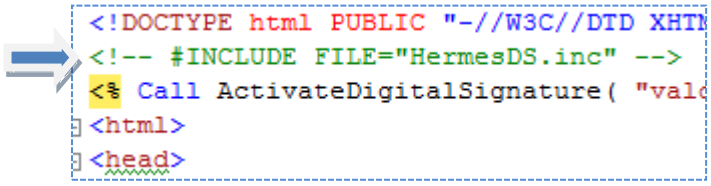
- Por último es necesario reiniciar el iis (iisreset)

Guía de integración con ASP

En esta sección se describe como integrar una aplicación ASP para firmar digitalmente las los formularios y validar la firma digital del lado del servidor.

Archivo HermesDS.inc

Este archivo debe ser incluido en las páginas ASP en las que se requiere activar la generación o validación de firma digital:



En HermesDS.inc se definen varias constantes cuyos valores deben definirse cuidadosamente para asegurar una correcta integración con el componente de firma digital:

CAId

Este parámetro numérico configura la autoridad certificadora que será utilizada por el componente de firma digital en el Internet Explorer. Los posibles valores son:

- *1004* para seleccionar la autoridad certificadora de prueba del SINPE
- *1006* para seleccionar la autoridad certificadora de producción del SINPE

IssuerDN

Este parámetro alfanumérico configura la autoridad certificadora que será utilizada por el componente de firma digital en Firefox. Los posibles valores son:

- "*CN=PRUEBAS - CA EMISORA PERSONA FISICA, OU=SINPE, O=BCCR, C=CR, serialNumber=CPJ-9-999-999999*" para seleccionar la autoridad certificadora de prueba del SINPE
- "*CN=CA SINPE - PERSONA FISICA,OU=DIVISION DE SERVICIOS FINANCIEROS, O=BANCO CENTRAL DE COSTA RICA,C=CR,serialNumber=4-000-004017*" para seleccionar la autoridad certificadora de producción del SINPE

GlobalDigSigSwitch

Este parámetro booleano activa o desactiva de manera global el uso de firma digital en todas las páginas. Los posibles valores de este parámetro son:

- *true* para activar la firma digital
- *false* para desactivar globalmente la firma digital

LocalDigSigSwitch

Este parámetro no debe ser modificado manualmente y se utiliza para activar o desactivar la firma digital para la página actual.

- *true* para activar la firma digital en esta página
- *false* para desactivar la firma digital en esta página

WebserviceURL

Este parámetro define la dirección del WSDL del servicio web de validación de firma digital, por ejemplo:

"http://192.168.1.7/fdg/fdg/webservices/ServicioFirmaDigital.asmx?**WSDL**"

Nótese que el valor de este parámetro debe terminar en “?WSDL”

IgnoreCRLFalsePositive

Este parámetro se debe utilizar solo para pruebas y controla si se acepta como valido una certificado cuya revocación no se pudo verificar.

- *true* acepta como valido un certificado cuya no revocación no se pudo validar.
- false nunca acepta como valido un certificado cuya no revocación no se pudo validar.

Generalmente este parámetro debe ser *false* pero se puede poner en *true* en un ambiente de prueba para aceptar firmas digitales realizadas con certificados cuya no revocación no se puede validar, por ejemplo porque el crl con que se cuenta es muy válido y tampoco se cuenta con un servicio OCSP.

Integración del componente de firma digital

En esta sección se describen los cambios que se deben realizar en una página ASP para lograr que el formulario sea firmado digitalmente antes de que este sea enviado al servidor.

El siguiente fragmento de código muestra una página ASP que ha sido integrada con la generación de firma digital:

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/
2  <!-- #INCLUDE FILE="HermesDS.inc" -->
3  <% Call ActivateDigitalSignature( "valor1", "valor1", "noimporta" ) %>
4  <html>
5  <head>
6      <meta http-equiv="content-type" content="text/html;charset=iso-8859-1" />
7  <% Call GenerateDigSigJSReference %>
8  </head>
9  <body>
10 <form action="ASPTest.asp" method="post" name="form1" id="form1"
11     onsubmit="return CallSignForm('este es el texto que se firmará');">
12 Your name: <input type="text" name="fname" size="20" />
13 <input type="submit" value="Submit"/>
14 <% Call GenerateSignatureField %>
15 </form>
16 <% Call GenerateCallSignForm %>
17 <% Call GenerateDigSigActivexReference %>
```

Ilustración 5 Código ASP integrado con firma digital

Las fechas rojas señalan las líneas de código ASP que fueron incluidas para generar el código html y javascript de soporte necesario para firmar digitalmente el formulario. A continuación se describen los cambios necesarios en una página ASP

Incluir HermesDS.inc (línea 2)

En la línea 2 se observa cómo se debe incluir el archivo HermesDS.inc. Este archivo contiene las funciones para la generación del código javascript y html necesario para generar y validar la firma digital del formulario.

Llamar a función de activación de firma digital (línea 3)

Esta función es responsable de activar o desactivar la firma digital en una página y debe llamarse desde el principio de la página como se muestra en el ejemplo. La implementación de esta función debe ser definida por el cliente siguiendo la siguiente guía:

- La función recibe tres parámetros genéricos cuyo significado es definido por el cliente, por ejemplo: id del cliente, número de cuenta y código del servicio.
- En base a esos parámetros la implementación de la función debe decidir si se activa o no la firma digital
- Si la función decide que se debe activar la firma digital entonces debe poner el valor de la variable global *LocalDigSigSwitch* en *true*
- Por el contrario si la función decide que no se debe activar la firma digital entonces debe poner el valor de la variable global *LocalDigSigSwitch* en *false*

Nótese que si *GlobalDigSigSwitch* es *false* entonces la firma digital estará deshabilitada de manera global y aunque *LocalDigSigSwitch* sea *true* no se activará la firma digital.

Incluir referencia a script "digsig.js" (línea 7)

La invocación a la función *GenerateDigSigJSReference* debe hacerse dentro del elemento *head* de la página y se utiliza para generar una referencia al script *digsig.js* como se muestra en la siguiente imagen:

```
<head>
    <meta http-equiv="content-type" content="text/html;charset=iso-8859-1" />
<script language="JavaScript" type="text/javascript" src="/js/digsig.js">s</script>

</head>
```

Ilustración 6 Referencia generada a digsig.js

El script digsig.js contiene varias funciones en javascript necesarias para firmar digitalmente.

Nótese que si la firma digital no está activa entonces *GenerateDigSigJSReference* no generará una referencia al script *digsig.js*.

Firmar digitalmente en evento onSubmit (línea 11)

Debe modificarse el elemento *form* de modo que el evento *onsubmit* llame a la función *javascript CallSignForm* como se muestra en la siguiente imagen:

```
<form action="ASPTest.asp" method="post" name="form1" id="form1"
    onsubmit="return CallSignForm('este es el texto que se firmará');">
```

Ilustración 7 Invocar función para firmar digitalmente

El parámetro que se pasa a la función *CallSignForm* es el texto de la transacción que se firmará digitalmente y que el usuario observará en el dialogo de confirmación para firmar digitalmente.

Nótese que si la firma digital no está activa entonces el llamado a *CallSignForm* no tendrá ningún efecto y el formulario se enviará normalmente sin firmar digitalmente.

Si se desea capturar el html del formulario que se va a firmar se deben seguir los siguientes pasos:

1. Se deben incluir en la página referencias a los scripts debug.js y dhtml.js de HAPI
2. Se debe llamar el método SaveFormBodyForHAF(...) antes de llamar a CallSignForm
3. El único parámetro que recibe SaveFormBodyForHAF(...) es el nombre de la página del action del form, por ejemplo SaveFormBodyForHAF(“demo.asp”)

Generar campo para guardar firma digital (línea 14)

Debe realizarse un llamado a la subrutina *GenerateSignatureField* en cualquier lugar dentro el elemento *form*. Este llamado sirve para generar el campo oculto *_DigSigNET1* el cual contendrá el valor de la firma digital. El campo generado se ve como el siguiente:

```
<input type="hidden" name="_DigSigNET1" id="_DigSigNET1" value="" />
```

Ilustración 8 Campo generado para contener firma digital

Nótese que si la firma digital no está activa entonces el llamado a *GenerateSignatureField* no tendrá ningún efecto y el campo *_DigSigNET1* no se generará.

Generar función para firmar digitalmente (línea 16)

Después del elemento *form* debe llamarse a la subrutina *GenerateCallSignForm* para generar la función de *javascript* que será llamada por el evento *onSubmit* del *form* y que se encargará de invocar las funciones *javascript* para generar la firma digital. El código generado es como el siguiente:

```
<script language="JavaScript" type="text/javascript">
function CallSignForm(msg)
{
        return SignFormPKCS7(null,false,msg);
}
</script>
```

Ilustración 9 Función generada para firmar digitalmente

Nótese que si la firma digital no está activa entonces el llamado a *GenerateCallSignForm* generará una función que siempre retorna true, de manera que no se generará la firma digital y el formulario se enviará normalmente.

Generar código de componente activex de firma digital (línea 17)

Al final de la página debe llamarse la subrutina *GenerateDigSigActivexReference* para incluir el código del componente *activex* de firma digital necesario para el Internet Explorer. El código generado por está subrutina es como el que se muestra en la siguiente imagen:

```
<object classid="clsid:0AE66022-15D3-4EB8-95ED-06E0E5CC484D"
    height="25" id="DigSigNET1" width="160"
    codebase="/fdg/DigSigSetup.exe#VERSION=1,15,0,0">
    <param name="ForeColor" value="16711680"></param>
    <param name="BackColor" value="16777215"></param>
    <param name="DateFormat" value="dd/mm/yyyy"></param>
    <param name="TimeFormat" value="hh:mm:ss"></param>
    <param name="IntCAID" value="1004"></param>
    <param name="ExtCAID" value="1004"></param>
    <param name="CAPIStore" value="MY"></param>

</object>
<script type="text/javascript" language="javascript">
    InitDigSigPKCS7('DigSigNET1', true, true, 'Arial', '8',
    '30/08/2009', '05:48:12',
    'CN=PRUEBAS - CA EMISORA PERSONA FISICA,OU=SINPE,O=BCCR,C=CR,
</script>
```

Ilustración 10 Referencia generada a Activex de firma digital

Nótese que si la firma digital no está activa entonces el llamado a *GenerateDigSigActivexReference* no generará ningún código.

Validación de la firma digital

El siguiente código ASP muestra como se valida la firma digital

```
dim signature, idCorrelacion
signature = Request.Form("_DigSigNET1")
Response.Write(signature)
If signature <> "" Then
    Dim result, errorMsg, cn, ssn, respOCSP, sn
    result = ValidateDigitalSignature( signature, errorMsg, cn, ssn, respOCSP, sn, "TRA", "CPF-01-0001-0002", idCorrelacion )
    Response.Write "<br/>Result=" & result & "<br/>"
    Response.Write "errorMsg=" & errorMsg & "<br/>"
    Response.Write "cn=" & cn & "<br/>"
    Response.Write "ssn=" & ssn & "<br/>"
    Response.Write "respOCSP=" & respOCSP & "<br/>"
    Response.Write "sn=" & sn & "<br/>"
    Response.Write "idCorrelacion=" & idCorrelacion & "<br/>"
```

Ilustración 11 Código ASP para validar una firma digital

El valor de la firma digital se obtiene del valor del campo *_DigSigNET1* y la función *ValidateDigitalSignature* se utiliza para validar esa firma digital. Esta función se encuentran en HermsDS.inc, recibe como primer parámetro la firma digital y retorna true si la firma digital fue validada positivamente o false si la firma digital fue validada negativamente o se produjo algún error técnico al tratar de realizar la validación. Los demás parámetros se utilizan para retornar información adicional de la firma digital:

- *errorMsg* retorna una descripción del error producido si la validación de la firma digital no es positiva
- *cn* retorna el campo *CommonName* del certificado usado para firmar
- *ssn* retorna el campo *SerialNumber* del certificado usado para firmar. Este campo contiene la cédula de identidad del titular del certificado.
- *respOCSP* retorna la respuesta del servicio OCSP codificada en base64
- *sn* retorna el número de serie del certificado usada para firmar digitalmente
- El parámetro *tipoTrans* que en este caso es “TRA” es un código de tipo de transacción que se puede definir en el mantenimiento de la bitácora de firma digital
- El parámetro *cedula* que en este caso es “CPF-01-0001-0002” contiene la cédula esperada del dueño del certificado usado para firmar digitalmente. Si la cédula del certificado no corresponde con el valor de este parámetro entonces se produce un error.
- El parámetro *idCorrelacion* se utiliza para retornar un identificador de la transacción que puede ser usado para referenciar la transacción firmada digitalmente en la bitácora, por ejemplo se necesita al llamar a la función *GenerateSignAndSavePDF* usada para generar un comprobante firmado digitalmente.

Firma digital de agente electrónico

La aplicación de firma digital de Hermes permite la generación de comprobantes o recibos firmados digitalmente utilizando certificados de agente electrónico. Estos comprobantes se generan en formato PDF firmado digitalmente utilizando el certificado de agente electrónico definido en el archivo de configuración. Para generar el PDF se utilizan plantillas en formato xsl-fo las cuales se almacenan en un directorio especial del servidor de aplicaciones.

El proceso para producir un comprobante firmado digitalmente se divide en dos pasos: la generación del comprobante firmado digitalmente y la descarga del comprobante. Las funciones necesarias para generar el comprobante y descargarlo se encuentran en el archivo HermesDS.inc

Generación del comprobante

La función que se utiliza para generar el comprobante firmado digitalmente se llama "*GenerateSignAndSavePDF*". El uso de esta función se muestra en la siguiente figura:

```
Dim xml
xml = "<Comprobante>"
xml = xml & "<Titulo>Comprobante de transferencia de tarjeta a cuenta</Titulo>"
xml = xml & "<Tabla>"
xml = xml & "   <Fila Encabezado=""Número de Tarjeta :"" Texto=""1234567890123456"" Alineamiento=""left"" />"
xml = xml & "   <Fila Encabezado=""Monto :"" Texto=""1000 colones"" Alineamiento=""left"" />"
xml = xml & "   <Fila Encabezado=""Número de Cuenta :"" Texto=""1234-5678"" Alineamiento=""left"" />"
xml = xml & "   <Fila Encabezado=""Fecha :"" Texto=""28/09/2009 10:30:00 AM"" Alineamiento=""left"" />"
xml = xml & "   <Fila Encabezado=""Código de Control :"" Texto=""987654"" Alineamiento=""center"" />"
xml = xml & "</Tabla>"
xml = xml & "</Comprobante>"

Dim comprobanteId
result = GenerateSignAndSavePDF( xml, "comprobante-fo.xsl", "Prueba firma digital", "Costa Rica", ssn,
    idCorrelacion, comprobanteId, errorMsg )
Response.Write "result=" & result & "<br/>"
Response.Write "comprobanteId=" & comprobanteId & "<br/>"
Response.Write "errorMsg=" & errorMsg & "<br/>"
```

Ilustración 12 Listado de código para generar comprobante

Esta función retorna "true" si se logró generar el PDF firmado digitalmente y "false" en caso de que se haya producido un error.

Los parámetros de esta función son los siguientes:

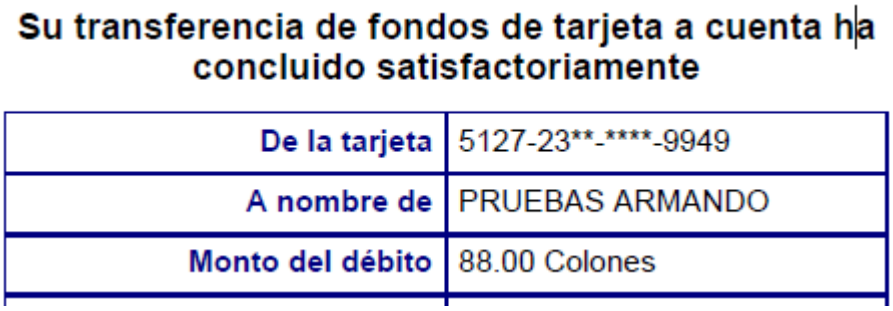
- xml. Este es el xml con los datos para generar el comprobante. Más adelante se explica la estructura de este xml.
- xslFilename. Es el nombre de la plantilla xsl que se debe utilizar para generar el comprobante. No debe incluir la ruta completa, solo el nombre del archivo. Por ahora la única plantilla definida es "comprobante-fo.xsl".
- reason. Texto libre con una descripción corta de la transacción.
- location. Lugar desde el que se firma, típicamente "Costa Rica"
- receiptSerialNumber. Número de cédula de la persona receptora del comprobante conforme al formato definido por el MICIT para las cédulas de los certificados. En el ejemplo se utiliza el número de cédula obtenido del certificado al llamar a la función *ValidateDigitalSignature*
- idCorrelacion. Id de correlación de la transacción a la cual está asociado este comprobante. Este debe ser el mismo IdCorrelacion que retorna la función *ValidateDigitalSignature*

- ComprobanteId. Id del comprobante generado. Se necesita posteriormente para descargar el PDF.
- errorMsg. En caso de un error en este parámetro se retorna una descripción del error que se produjo.

Formato de xml para comprobante

El primer parámetro que se pasa a la función *GenerateSignAndSavePDF* es un xml con los datos que se desplegarán en el comprobante firmado digitalmente. El formato de este xml dependerá de la plantilla xsl que se utilice para generar el comprobante. Actualmente la única plantilla xsl definida es “comprobante-fo.xsl”.

La plantilla “comprobante-fo.xsl” muestra una tabla donde la primer columna es el encabezado de la fila y la segunda columna es el correspondiente valor. El siguiente es un fragmente de un comprobante generado con esta plantilla.



Su transferencia de fondos de tarjeta a cuenta ha concluido satisfactoriamente

De la tarjeta	5127-23**-****-9949
A nombre de	PRUEBAS ARMANDO
Monto del débito	88.00 Colones

Ilustración 13 Fragmento de comprobante generado con comprobante-fo.xsl

El siguiente es un ejemplo del XML requerido para la plantilla “comprobante-fo.xsl”:

```
Dim xml
xml = "<Comprobante>"
xml = xml & "<Titulo>Comprobante de transferencia de tarjeta a cuenta</Titulo>"
xml = xml & "<Tabla>"
xml = xml & "   <Fila Encabezado=""Número de Tarjeta :"" Texto=""1234567890123456"" Alineamiento=""left"" />"
xml = xml & "   <Fila Encabezado=""Monto :"" Texto=""1000 colones"" Alineamiento=""left"" />"
xml = xml & "   <Fila Encabezado=""Número de Cuenta :"" Texto=""1234-5678"" Alineamiento=""left"" />"
xml = xml & "   <Fila Encabezado=""Fecha :"" Texto=""28/09/2009 10:30:00 AM"" Alineamiento=""left"" />"
xml = xml & "   <Fila Encabezado=""Código de Control :"" Texto=""987654"" Alineamiento=""center"" />"
xml = xml & "</Tabla>"
xml = xml & "</Comprobante>"
```

Ilustración 14 Formato XML esperado por comprobante-fo.xsl

- Elemento “Titulo”. El texto contenido en este elemento se utiliza para el titulo del comprobante.

- Elemento “Fila”. Cada elemento de este tipo representa una fila del comprobante compuesta por un encabezado de la fila y el respectivo valor
- Atributo “Encabezado” del elemento “Fila”. Es el encabezado de la fila.
- Atributo “Valor” del elemento “Fila”. Es el valor que se muestra en la segunda columna
- Atributo “Alineamiento” del elemento “Fila”. Es el alineamiento de la celda de valor. Los posibles valores son: left, center y right.

Descarga del comprobante mediante llamado a función

Luego que se generó el comprobante el siguiente paso es realizar la descarga para que el destinatario pueda obtenerlo. La función que se utiliza para obtener el PDF es GetPDF. El siguiente es un ejemplo del uso de esta función:

```
Dim receiptSerialNumber, ComprobanteId, pdf, errorMsg, result

receiptSerialNumber = Request.Form("ssn")
ComprobanteId = Request.Form("comprobanteId")
result = GetPDF( receiptSerialNumber, ComprobanteId, pdf, errorMsg )
If result Then
    Response.Clear
    Response.ContentType = "application/pdf"
    Response.AddHeader "content-disposition", "attachment; filename=""recibo" & ComprobanteId & ".pdf"""
    Response.BinaryWrite( pdf )
    Response.End
Else
    Response.ContentType = "text/html;charset=iso-8859-1"
    Response.CharSet = "iso-8859-1"
    Response.Write( errorMsg )
End If
```

Ilustración 15 Ejemplo de uso de función GetPDF

GetPDF retorna “true” si se logró obtener el comprobante exitosamente o “false” si se produjo un error.

Los parámetros de GetPDF son:

- receiptSerialNumber: Número de cédula de la persona receptora del comprobante conforme al formato definido por el MICIT para las cédulas de los certificados.
- ComprobanteId. Id del comprobante retornado por el llamado a la función *GenerateSignAndSavePDF*
- pdf. Si todo sale bien en este parámetro se retorna el pdf como un array de bytes
- errorMsg. Si se produce algún error en GetPDF en este parámetro se retorna la descripción del mensaje de error.

Descarga del comprobante mediante DescargarComprobante.asp

La página DescargarComprobante.asp ya implementa la funcionalidad para descargar el comprobante. La forma de utilizar está página es como se muestra en el siguiente fragmento de código:

```
Response.Write "<form action=""DescargarComprobante.asp"" method=""post"">" & vbCrLf
Response.Write "<input type=""submit"" value=""Descargar Recibo""/>"  & vbCrLf
Response.Write "<input type=""hidden"" name=""comprobanteId"" value=""" & comprobanteId & """/>" & vbCrLf
Response.Write "<input type=""hidden"" name=""ssn"" value=""" & ssn & """/>" & vbCrLf
Response.Write "</form>"
```

Ilustración 16 Ejemplo de invocación a página DescargarComprobante.asp

La página DescargarComprobante.asp espera los siguientes parámetros:

- ComprobanteId. Id del comprobante retornado por el llamado a la función *GenerateSignAndSavePDF*
- ssn. Número de cédula de la persona receptora del comprobante conforme al formato definido por el MICIT para las cédulas de los certificados.

Instalación de un certificado de agente electrónico

En esta sección se describe como instalar un certificado digital de agente electrónico en Windows de manera que el servicio de validación de firma digita lo pueda acceder para generar los comprobantes firmados digitalmente

A continuación se explica cómo:

- Instalar el certificado en el store de certificados de la computadora local
- Poner los permisos al certificado para poder accederlo con el usuario “Network Service”

Instalar certificado en el store de certificados de la computadora local

Pasos para importar el certificado en el store de certificados de la computadora personal:

1. Ejecutar la aplicación mmc
2. Clic en File | Add / Remove Snap-in
3. Seleccionar Certificates en la lista de la izquierda, hacer clic en el botón “Add >”
4. Seleccionar “Computer account” en el dialogo “Certificates snap-in”, seleccionar “Local Computer” en el dialogo “Select computer” y clic en el botón Finish
5. Clic en el botón “OK” del dialogo “Add/Remove snap-in”
6. Expandir el nodo “Certificates”, expandir el nodo “Personal”, clic derecho en el nodo “Certificates”, clic en “All Tasks”, clic en “Import...”

7. Clic en el botón “Next” del dialogo “Certificate import wizard”, seleccionar el archivo pfx o p12 que contiene el certificado a importar, clic en el botón “Next”, introducir la clave del pfx, clic next en el siguiente paso, y clic “Finish” en el paso final.

Pasos para permitir que el usuario “Network service” acceda el certificado

1. Instalar la aplicación winhttpcertcfg
   El instalador se encuentra en la ruta \fdg\winhttpcertcfg.exe del CD de instalación
2. Ejecutar el comando winhttpcertcfg -g -c LOCAL_MACHINE\MY -s xxxx -a "Network Service"

Notese que xxxx debe reemplazarse con el valor del componente CN (common name) del campo subject del certificado al cual se le quiere dar acceso al usuario “Network Service”

Configuración del firmador

El firmador de documentos de HAPI debe configurarse para que pueda instarse desde el servidor web del cliente y para que se comunique con el servicio de validación del cliente. Para hacer esto son necesarios dos pasos:

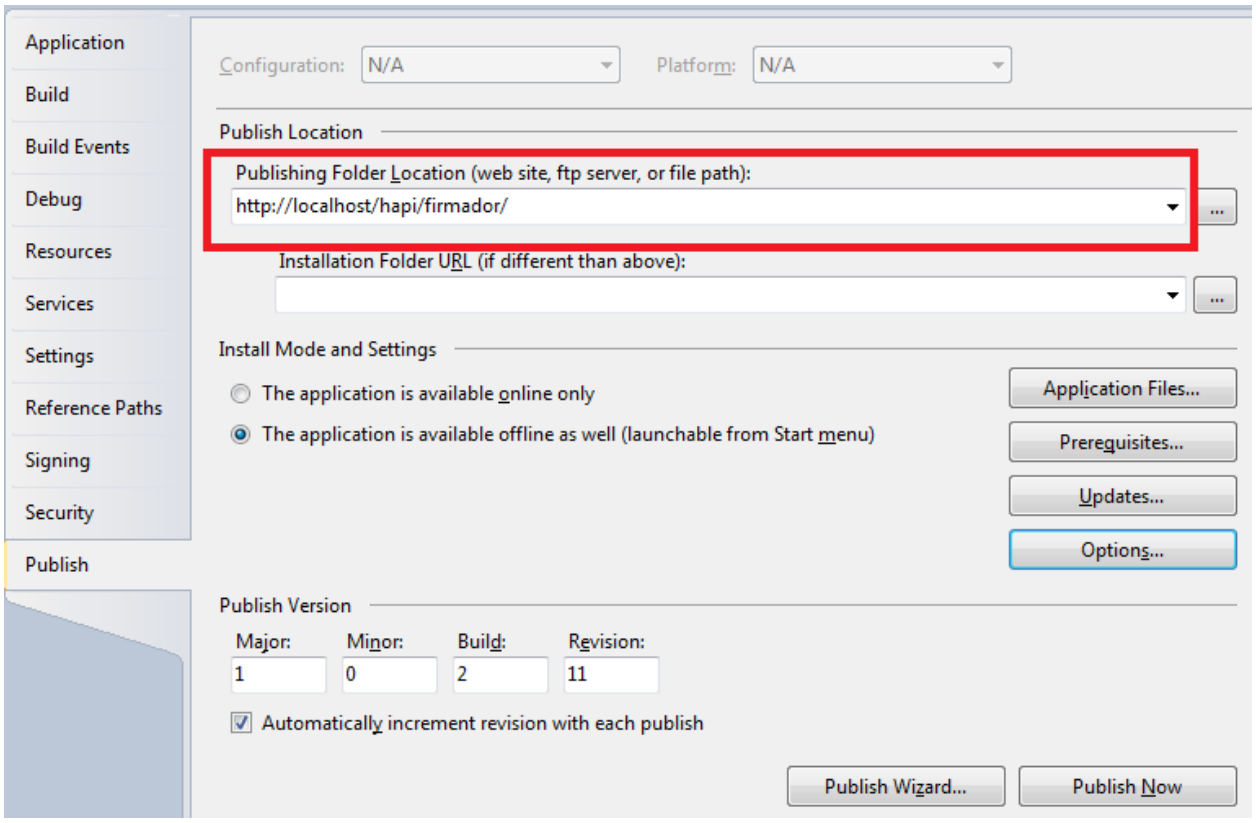
- Configurar la dirección del webservice de validación en la configuración del firmador
- Configurar la dirección de instalación del firmador y generar el instalador

Configurar la dirección del webservice de validación

Este paso es necesario para que el Firmador de HAPI se pueda conectar al servicio de validación del cliente. Se debe modificar los valores de las propiedades “address” de los elementos configuration/system.serviceModel/client/endpoint del archivo app.config del proyecto FirmadorHAPI. Por ejemplo un posible valor sería http://www.hermes-soft.com/hapi/fdg/WebServices/ServicioFirmaDigital.asmx

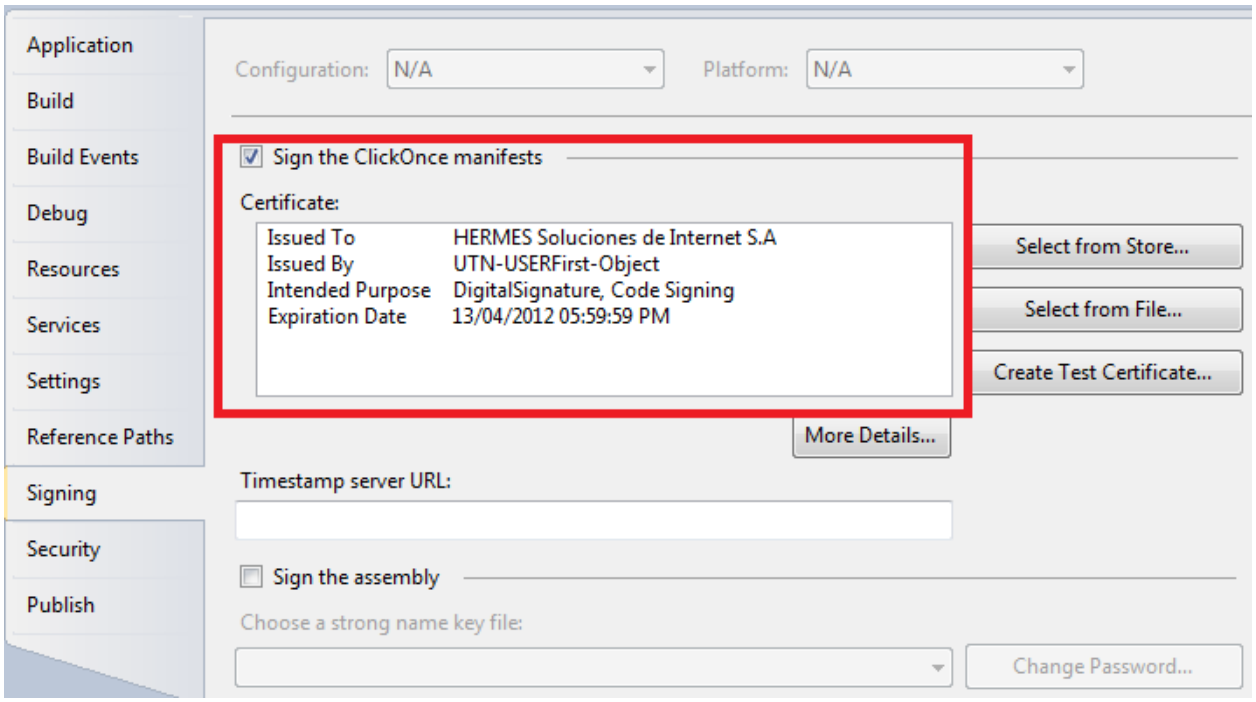
Configurar la dirección de instalación del firmador

Este paso se tiene que hacer desde Visual Studio. Clic derecho en el proyecto “Firmador HAPI”, Clic en la opción “Propiedades” y clic en la pestaña “Publish”:



Debe modificarse el campo “Installation folder URL” para que apunte a la dirección del cliente, por ejemplo “http://www.hermes-soft.com/hapi/firmador.

Además debe verficarse que se firme digitalmente el manifiesto click-once en la pestaña “Signing”:



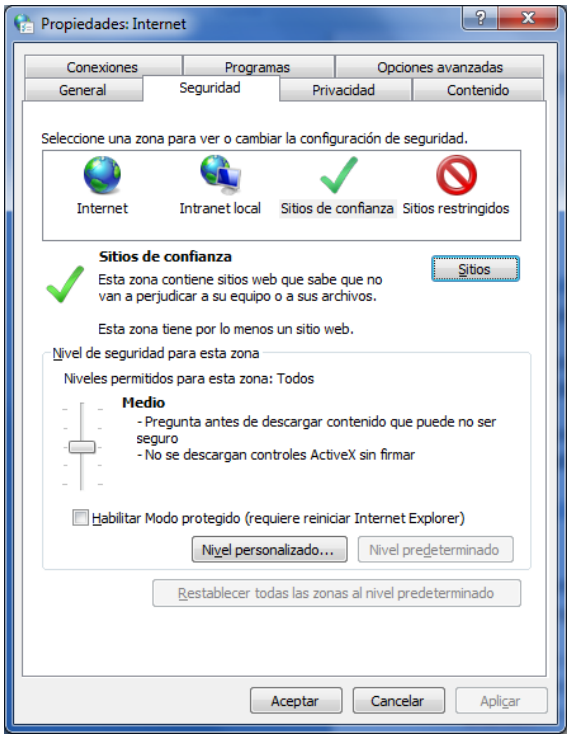
Se debe hacer clic en el botón “Publish Now” de la pestaña Publish. Una vez que se genera el instalador debe copiarse la carpeta “D:\HermesDSApp\Demos\Web\firmador” en el directorio “hapi” del sitio del cliente.

Configuración de las computadoras de los usuarios

Agregar HAPI a los sitios de confianza

Para mejorar la compatibilidad con el sitio y evitar algunos inconvenientes, se debe incluir la dirección del sitio de HAPI dentro de los Sitios de confianza del Internet Explorer.

Para ello se accede al **Panel del Control,** selecciona **Opciones de Internet** y activa la pestaña **Seguridad**.



Una vez ahí, selecciona **Sitios de confianza** y presiona el botón **Sitios**. Se abrirá una pequeña ventana en la cual debe agregar la url del dominio de HAPI.

Deshabilitar bloqueadores de ventanas emergentes (popups)

Para el correcto funcionamiento del sistema, se deben deshabilitar los bloqueadores de popups en las computadoras de los usuarios del sistema.

En el caso de Internet Explorer 7 o superior, éste le consulta al usuario por medio de la barra amarilla de alerta, se debe hacer clic en la barra y seleccionar la opción: **Siempre permitir popups de este sitio**.

Si se utilizan bloqueadores externos como por ejemplo la barra de Google o el Ad-block Plus para Firefox, se deben deshabilitar para el sitio de HAPI.