User’s Guide to IRP Client v0.8

Lawrence E. Hughes, CTO
Sixscape Communications, Ltd.
22 March 2014

IRP (Identity Registration Protocol) is similar to DNS (Domain Name System) in some respects. Unlike DNS it allows registration of information related to a *person* (including their current IPv6 address), not information specific to some *device*. It also includes a full featured PKI (Public Key Infrastructure). It is better suited to on-the-fly authenticated registration of people’s identity from highly mobile nodes, compared with DNS which was designed for primarily manual registration of largely static nodes (mostly servers, in the case of “external DNS”).

The IRP system consists of several components:

- An *IRP Server* (one for each Internet domain). This is currently deployed as a Windows Service (similar to a UNIX daemon) written in C#.Net. IRP clients connect to the *IRP Server* via a simple XML based client/server protocol, over IPv4 or IPv6. Only the Server and the Admin tool access the database directly. Clients access the database only indirectly through the server.

- A database used by the *IRP server*. For small sites (up to a few thousand users), this can be easily handled with MS SQL Server Express. Larger sites can deploy the database on a large MS SQL Server engine. On the Linux version the database will be implemented with PostgreSQL.

- An *IRP Admin Tool* for handling manual administrative tasks, such as creating the root and intermediate keys and certificates for a certificate hierarchy, administrative revocation of certificates, etc. Currently this is a Windows GUI app, written in C#.Net. This tool can easily access any compatible SQL Server, on the same box or remote.

- Various IRP-enabled clients that use the IRP server.

    - The simplest IRP enabled client is the basic *IRP Client* that allows a user to request and download a client digital certificate, and install it in their Microsoft Certificate Store (part of IE, also used by Outlook, Windows Live Mail, etc). Optionally the downloaded keys and certs and be installed in the Mozilla certificate database (part of Firefox, also used by Thunderbird – to be supported later). This is sufficient for web based strong client authentication, S/MIME email with existing clients such as Outlook, and strong authentication to a network OS, such as Windows Server. This will be available for Windows, Android and iOS (currently only Windows 7).

    - A basic GUI based email client can be created that has IRP support built-in. However, many people are really tied to their existing email client (e.g. Outlook), so we will also provide a plug-in for Outlook that will obtain a client cert from IRP, install it in the Microsoft Certificate Store and configure Outlook to use it. It will also be able to obtain recipients' certificates for sending encrypted messages from IRP, at the time the message is composed. A mobile based email client with similar capabilities will be created for Android and iOS, again with IRP support built in.

- The *SixChat* P2P secure messaging system uses IRP both for obtaining client certs, as a directory, and as a way to register its IPv6 address and obtain those of other users. A SixChat User Agent will obtain and install a client cert (and associated CA certs), and publish those and other information in the IRP directory, together with the agent's current IPv6 global address. Other SixChat User Agents will be able to locate your SixChat client via the IRP server. When you connect, the SixChat User Agents will exchange client certs and a symmetric session key (P2P TLS) so that the remainder of the call for strong mutual authentication and End-to-End privacy. SixChat will also be available for Windows, Android and iOS (maybe later Mac OS-X and/or Linux/GNOME). SixChat includes Chat, P2P Mail and P2P file transfer, all with strong security, over IPv6. A legacy mode will allow either or both ends to be over IPv4, using IRP servers as relay nodes between the public IPv4 Internet and a given private IPv4 Internet.

- The *SixPhone* P2P VoIP softphone will use IRP similar to the SixChat system for real P2P voice over IPv6, in addition to being to work in a Client/Server model for use with legacy (IPv4 client/server) VoIP products. The SIP protocol will be extended to provide strong mutual authentication and encryption similar to SixChat. This will be available for the same platforms as SixChat.

- A *Secure File Repository* can be created similar to an FTP or SSH server (e.g. sshd from Linux), but with a new protocol and built-in support for IRP. This server can be created for both Windows (as a service) and Linux (as a daemon). It will use TLS and strong authentication for the connection, and allow S/MIME encryption and digital signatures of files both during transmission to and from the Repository and in storage in the Repository. For example, a file could be stored encrypted for retrieval by any number of recipients using certs obtained from IRP. The users could obtain their own certs and private keys, and cert of other users via the basic IRP client. The file sending and receiving functions could be integrated into the IRP client.

IRP Client

The *IRP Client* is a Windows based app, written in C#.Net. It is a standalone tool that runs on Windows 7 or Windows Server 2008+, and is used to obtain directory and address information, as well as a client digital certificate from the IRP Server. It works with Client digital certificates that “bind” a user’s name, UserID and email address to a public key. *Binding* is accomplished by digitally signing a CSR (Certificate Signing Request) that contains both a public key and the identifying items. The IRP Client is part of an automated PKI system (IRP –Identity Registration Protocol) that is accessed from clients via network protocols. The *IRP Admin Tool* (described elsewhere) is used for creating the root and intermediate certs of a hierarchy, viewing certs, approving and signing certs manually, etc. The IRP Admin tool can also be used to create CRLs (Certificate Revocation Lists) or configure an OCSP (Online Certificate Status Protocol) server. The IRP server implements IRP to allow IRP-enabled clients to register and obtain PKI and address information securely. It includes certificate revocation information (to replace CRLs and OCSP).

The IRP client can create a CSR and submit it to the local domain IRP server, then retrieve the generated certificate, over IPv4 or IPv6. It can also obtain certificate for any other IRP user, from their IRP server (based on their Internet domain name, found using DNS SRV records). This is useful for sending encrypted messages or objects to other users, without requiring them to provide their public key to you in advance. It can also install your client cert in the Microsoft certificate database (and obtain and install necessary root and intermediate certs) from the IRP server, making private hierarchy certificates practical.

The IRP client also automatically registers its current IPv6 address, and can obtain the IPv6 address of any other IRP user. This makes IPv6 End2End connectivity possible (connections directly from one IRP-enabled User Agent to another, compared with client/server architecture made necessary by NAT on IPv4). The PKI aspects of IRP make *Secure* End2End (and secure Client/Server) connectivity possible.

Finally the IRP client can encrypt or decrypt (as well as digitally sign and verify signatures) for any file on your device, using symmetric session keys exchanged using client certs (like S/MIME, but for any file, not embedded in an email client). This allows anyone to send secured files through facilities like DropBox, OneBox, etc. Since the format used by the IRP client in creating these secured files is IETF standard S/MIME, any email client software (Outlook, Windows Live Mail, Thunderbird, etc) is capable of loaded the secured files sent this way (they appear in your email as if they arrived via SMTP). Once in your email InBox, attachments can be saved as usual.

Types of Certificates

Server Cert – the public key is bound to a Fully Qualified Domain name (e.g. www.sixscape.com). Server certs are used on SSL/TLS servers to enable SSL/TLS by providing strong server to client authentication and symmetric session key exchange (for privacy). A single server certificate allows any number of clients to establish secured connections to a server, but only provides *server to client* authentication.

The ability to request and obtain Server certs can be easily added to IRP. These are private hierarchy, and so are not suitable for general public website use, but any IRP user can easily obtain the necessary root and intermediate (CA) certs required to trust these server certs. The CA certs need to be installed on both the web server and all client nodes that use it. You would typically create a CSR (Certificate Signing Request) for a server cert from an IRP client, submit it, then download the signed server cert. The key and cert would then be installed (along with the CA certs) on the web server (Apache, IIS, etc). Some server (e.g. IIS, Exchange) generate a CSR themselves. This can be saved to a file and submitted using the IRP client. These server certs are well suited for internal (intranet) servers, or for servers used only by your corporate customers. They are also suitable for securing local access to mail servers (e.g. IMAP, webmail), but since they are not public, they are not suitable for securing SMTP MTA to MTA transfers (this requires all SMTP MTAs to trust the server cert used).

Client Cert – the public key is bound to a person's name, email address and or UserID. Client certs are used for various purposes:

- S/MIME email
- Strong *client to server* authentication for any server running SSL/TLS that supports strong client authentication (web, email, LDAP, etc)
- Peer authentication and symmetric session key exchange in Peer-2-Peer SSL/TLS based User Agents (like SixChat)
- User authentication to a network OS (e.g. Windows Server), from a client PC
- Digital signing of XML files, PDF files and executable applications (code signing)

Unlike server certs, one client cert is needed for each user of a secure system.

IPsec Cert – public key is bound to one or more IP addresses. IPsec certs are used for mutual authentication during IKE (Internet Key Exchange) setting up IPsec (Internet Protocol Security) connections, either linking two networks or a remote user to a corporate network. IPsec certs are not used for key exchange (IKE uses Diffie-Hellman Key Agreement to exchange a symmetric encryption key). Usually IPsec certs are obtained from a CA (Certification Authority) via SCEP (Simple Certificate Enrollment Protocol). We can possibly add SCEP to the IRP server to allow existing network products to get IPsec certs from the IRP server. They can be managed easily via IRP.

Certificate Hierarchies

Except for *self-signed certs* (ones in which the public key cert is digitally signed by the corresponding private key), digital certificates are usually created in *hierarchies*. A hierarchy consists of a single root cert (the root of the hierarchy), one or more intermediate certs (forming a chain from the root cert down to the end-entity certs), and a potentially large number of end-entity certs, signed by the lowest CA cert in the hierarchy chain. In a two level hierarchy (root + end-entity certs) this would be the root cert. In a three level hierarchy (root cert + one intermediate cert + end-entity certs) this would be the intermediate cert. Each level mentioned here actually consists of both a public key digital certificate and the private key corresponding to the public key in that cert. So there are parallel hierarchies of certs and private keys. Signing of certs is done using the private keys.

In a three-level hierarchy, the root cert is self-signed (by the root private key). The intermediate cert is signed by the root private key. End-entity certs are signed by the intermediate private key. Certificate validation is done by checking the digital signature of an end-entity cert, using the intermediate cert, then checking the digital signature of the intermediate cert, using the root cert. The root cert must be verified manually and marked as trusted on the client or server. In the Microsoft certificate store (part of Internet Explorer), this is done by storing a root certificate in the *Trusted Root Certification Authorities* cert storage folder. Intermediate certs are usually stored in the *Intermediate Certification Authorities* folder. In the Mozilla certificate store (part of Firefox) this is done by storing a root cert in the *Authorities* folder and then marking it as *trusted* (for one or more purposes). Both Microsoft and Mozilla certificate stores have numerous trusted root certificates installed by default (these are for *public hierarchies*).

To be clear, a *public hierarchy* is one in which the root cert comes preinstalled in client and server software from the software vendor (and presumably verified as trustworthy by that vendor). These are updated periodically (e.g. via Windows Update). A *private hierarchy* is one where the root cert must be obtained and installed in all relying nodes. This is normally a complicated insecure process, but with IRP if you trust the domain server, you can trust (and easily download and install) its root and intermediate (CA) certs, on any relying node (client or server).

The IRP Client can store certs and private keys into the Microsoft certificate store, or import or use certs and private keys from it. With either Microsoft or Mozilla certificate store, you can install certs into the database by making the appropriate browser the default, then double clicking on the cert (in PEM, DER or PFX format).

Certificate Formats

CSR refers to a *Certificate Signing Request*. It is an encrypted object that contains a public key, identifying information and various other items. CSRs were originally defined in PKCS #10 (part of the RSA Public Key Cryptography Standards). They are now defined in RFC 2986, “PKCS #10: Certification Request Syntax Specification, v1.7” (part of the IETF PKIX group of standards, which stands for *Public Key Infrastructure with X.509*. You create a CSR and submit it to a *Certification Authority* (CA) who can create an X.509 digital certificate from it.

Certificate (or **Cert**) refers to a digital document that contains a public key (usually for the RSA public key algorithm), and various other identifying information. The entire document is digitally signed by a *Certification Authority* (CA), also known as a *Trusted Third Party*. It is a safe envelope for storing and/or exchanging public keys. They are useful only in conjunction with the private key corresponding to the public key in the certificate. The private key is usually kept protected by symmetric key encryption (e.g. 3DES or AES). You can share your public key with anyone. It can be used to encrypt something that only you can decrypt (because you are the only one that has your private key), or anyone can decrypt something you encrypted with your private key (since anyone can obtain your digital certificate).

The following file formats are used for import and export of certificates and private keys. The

PEM (Privacy Enhanced Mail): an ASCII text encoded file that can contain one or more CSRs, one or more private keys and/or one or more certs, optionally with encryption. The most common forms are:

- CSR stored by itself (no encryption)
- Private key by itself (with encryption)
- Cert stored by itself (no encryption)
- CSR + corresponding private key (with encryption)
- Cert + corresponding private key (with encryption)
- Two or more certs, typically from a single hierarchy, such as root and intermediate (without encryption)

DER (Distinguished Encoding Rules): a binary format (from X.509) that has no provision for encryption and passwords. The most common forms are:

- CSR stored by itself (no encryption)
- Private key by itself (no encryption)
- Cert stored by itself (no encryption)
- Two or more certs, typically from a single hierarchy, such as root and intermediate (no encryption)

PKCS #12 (or **PFX**): an encrypted binary file format that can hold a private key, a cert or both (usually both). It can actually contain several certificates and/or keys, for example an end-entity cert and the CA certs required to check the end-entity cert’s validity. This is used for securely storing or transferring key material. The original PKCS #12 standard is available from RSA, but the IETF standard that incorporates it is RFC 5959, “Algorithms for Asymmetric Key Package Content Type”, August 2010.

PCKS #11 compliant devices (e.g. USB security tokens) can usually import keying material (certs and/or private keys) from PKCS #12 files. Both Microsoft and Mozilla software can import and export keying material in PKCS #12 format. This can be used for secure *backup and recovery* of keying material, including private keys, since only the person knowing the passphrase it was encrypted with can recover the keying material. This does not meet the requirements for *key escrow*, which allows the organization (or government) to retrieve keying material without the assistance (or even knowledge) of the key owner. It is possible for the CA to generate all public/private keys internally and provide keying material to end-users in PKCS #12 form for import to their software products. In that case, the CA has access to every user’s private keys. The most common filetype for PCKS #12 packages is *.pfx*, although *.p12* is also used. Sometimes PKCS #12 format is sometimes referred to as “PFX” format because of this.

IRP Client and *IRP Admin Tool* Functionality

The *IRP Client* can generate a public/private keypair, allow entry of identifying information (Subject's Distinguished Name) and produce a CSR plus private key. It can submit that CSR securely to an IRP server, where it can be approved and digitally signed, using the private key of that IRP server (technically the private key corresponding to the lowest level CA cert of the hierarchy used). You can then download the generated certificate (and if needed, the root and intermediate certificates of that server). You can also obtain the digital cert of any other IRP user. You can create a PKCS #12 package from your certificate and private key, using a passphrase known only to you, and securely upload it to the IRP Server, which keeps it in the IRP database for you. You can download this to restore your keying material if it is lost or destroyed, or install it on other devices. Nobody buy you can access the keying material (including the IRP administrator) if you don't disclose the passphrase you used when you created the PKCS #12 package.

It is also possible to use the IRP system with the *Key Escrow* model, in which your identifying information (Subject distinguished name) is entered by the IRP administrator using the *IRP Admin Tool* (not the IRP client). The *IRP Admin Tool* creates a keypair, then generates a certificate from the public key, and saves the key and certificate in a PKCS 12 package, which can be downloaded and installed in your certificate database (again, including necessary root and intermediate certs). In this model, the IRP administrator has access to every user's private keys. In this model, the IRP administrator (or HR department, etc) must securely relay the name of the PKCS #12 package and the passphrase needed to open it to the user. The IRP Client can be used to create an identity, and the PKCS #12 package can be downloaded and associated with that identity. This is much less secure than the usual model, where the keypair is created on the client node and the private key is never sent to the IRP server, except in a protected PKCS #12 package. Everybody's private keys are kept on the IRP server in this model. While they are kept encrypted with a symmetric key, it is possible a hacker could obtain this key and thereby everybody's private keys. This would allow the hacker to access everyone's encrypted files and assume their identity.

Because of this vulnerability, unless key escrow is required by the government or corporate policy, it is better to use the normal model (wherein keypairs are generated on the client node).

Optionally, an IRP client can create a CSR containing the distinguished name of a server (e.g. www.sixscape.com), submit it, and obtain a server certificate suitable for enabling SSL or TLS on a web server, email server, etc. This will not be a public hierarchy, but again, any user can obtain the necessary root and intermediate certs of that hierarchy and use the server without paying for a real public cert. This is suitable for intranet servers, or for customers of the organization running the IRP system. The client certs can be used for strong client authentication to these servers, as well. The IRP client user needs only to install the downloaded cert and private key into the server.

The *IRP Admin Tool* can create a self-signed root certificate and an intermediate cert, signed by the root cert. These establish a certificate hierarchy. The IRP Admin Tool can use an intermediate cert's private key to sign end-entity certs, from keying material stored in the IRP database, or obtained from clients in the form of CSRs. Ideally the root *private key* is kept offline in a highly secure storage vault, and is used only occasionally to create new intermediate certs (say once every few years). The root cert can have a lifetime of 10-30 years. Intermediate certs can have lifetimes of 5-10 years. End-entity certs usually have lifetimes of 1 year, but this could be set to anything based on the needs of the customer. Issued (but unexpired) certs can be revoked by advertising them in an OCSP (Online Certificate Status Protocol) server (IRP includes an OCSP server, and can produce and publish CRLS – it also allows clients to check revocation status via its own protocols.

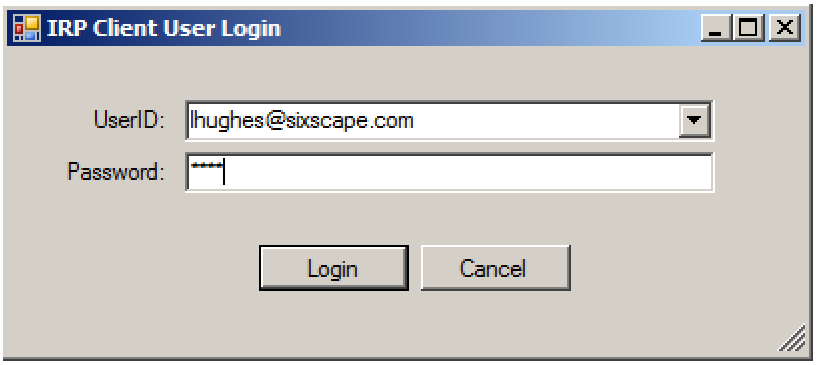
The *IRP Admin Tool* can also create deeper certificate hierarchies (more than one intermediate cert). IN this case, each cert is signed by the private key corresponding to the next higher level cert (ending with the self-signed root cert). The private key associated with the lowest level intermediate cert is used to sign end-entity certs. Relying nodes must have the root and all intermediate certs installed before end-entity certs from that hierarchy can be validated.

All clients should validate every cert they use, including checking their revocation status (using OCSP, CRLs or IRP's certificate revocation messages) before accepting them as valid for authentication or symmetric key exchange. The IRP Client can validate certificates using CRLs, OCSP or IRP. Any IRP-enabled client (e.g. Outlook with an IRP plug-in) can do the same thing.

The *IRP Admin Tool* accesses the IRP database directly, so it must run on a system that has access to that database, often the computer where the database resides (although it can access a SQL Server database on another computer, if desired). The IRP server also interacts with the IRP database, and must also be able to access that database. An IRP client never accesses the IRP database directly, only via the *IRP Server*. The IRP client never accesses the *IRP Admin Tool* – only the IRP Server (potentially any IRP server, to obtain certificates of users in other domains).

IRP Client – Login

When you first execute the IRP client, it will challenge you for a user login.



UserID: lhughes@sixscape.com

Password: ****

The available current Personal Identities (created or downloaded on your node) will be listed in the UserID GUI control. Each Personal Identity is store in an .xml file in the *IRP_Client/Personal* directory. The most recently logged in user will be shown (tracked in the Registry). If there is no recent user, the first Personal Identity found will be shown. If there are no Personal Identities yet, the UserID GUI control will be blank.

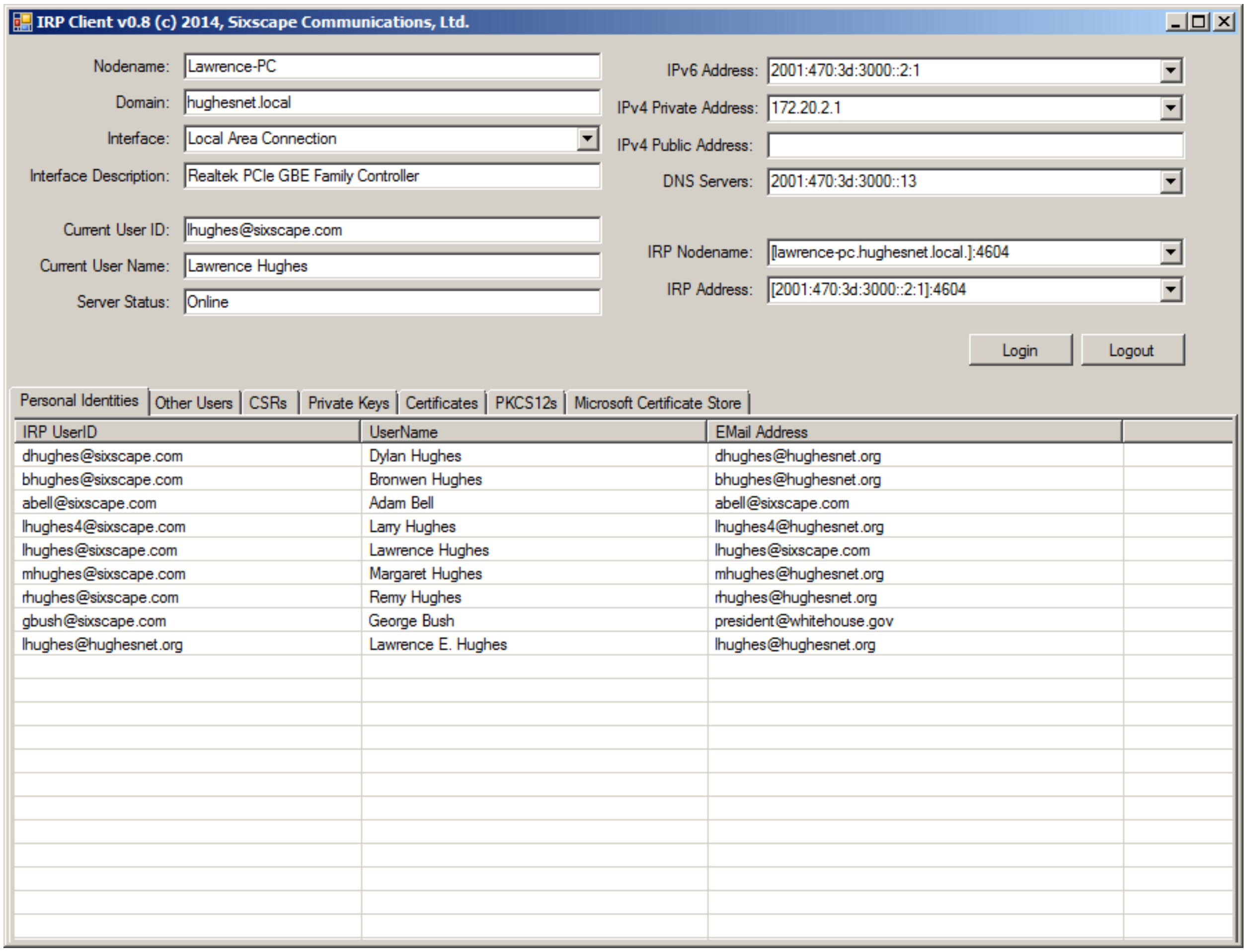
Assuming at least one Personal Identity exists, either accept the one shown, or click the combobox arrow and select another Personal Identity from the resulting pulldown list. Enter the password for that Identity (it will echo as "*" characters). Then click *Login* (or just hit Enter – *Login* is the default control). The IRP Client will send an *Authenticate_User* message to the IRP server. If you entered the correct password, it will advance to the main screen. If not, you have up to 3 tries. If you enter the wrong password three time (or click the *Cancel* button), the application will terminate.

If the Personal Identity xml file is not on your current machine (this could happen if this is the first time you have logged in on this machine), but you have a valid IRP account on your IRP server, enter your UserID manually in the *UserID* GUI control and the password in the *Password* GUI control, and then click *Login*. Assuming you have the correct password, this will authenticate you against the information on the server, and advance to the main screen logged in with your UserID. You should download your Personal Identity file from the server, so that in the future you can pick it off of the pull down menu.

The server password is also written (in hashed form) to the Personal Identity file, to allow you to login to the IRP client even if the server in not currently available. If you have not logged in at least once on this machine you will not be able to authenticate, unless you manually copy your Personal Identity file into the the *IRP_Client/Personal* directory.

[Coming soon: If you login and your Personal Identity file is not present, the IRP Client will automatically download it for you from the IRP Server.]

The main screen of the IRP Client looks like this:


IRP Client v0.8 (c) 2014, Sixscape Communications, Ltd.
Login
Logout
Personal Identities
Other Users
CSRs
Private Keys
Certificates
PKCS12s
Microsoft Certificate Store

Nodename: Lawrence-PC
Domain: hughesnet.local
Interface: Local Area Connection
Interface Description: Realtek PCIe GBE Family Controller
Current User ID: lhughes@sixscape.com
Current User Name: Lawrence Hughes
Server Status: Online

IPv6 Address: 2001:470:3d:3000::2:1
IPv4 Private Address: 172.20.2.1
IPv4 Public Address:
DNS Servers: 2001:470:3d:3000::13
IRP Nodename: [lawrence-pc.hughesnet.local.]:4604
IRP Address: [2001:470:3d:3000::2:1]:4604

IRP UserID	UserName	EMail Address
dhughes@sixscape.com	Dylan Hughes	dhughes@hughesnet.org
bhughes@sixscape.com	Bronwen Hughes	bhughes@hughesnet.org
abell@sixscape.com	Adam Bell	abell@sixscape.com
lhughes4@sixscape.com	Larry Hughes	lhughes4@hughesnet.org
lhughes@sixscape.com	Lawrence Hughes	lhughes@sixscape.com
mhughes@sixscape.com	Margaret Hughes	mhughes@hughesnet.org
rhughes@sixscape.com	Remy Hughes	rhughes@hughesnet.org
gbush@sixscape.com	George Bush	president@whitehouse.gov
lhughes@hughesnet.org	Lawrence E. Hughes	lhughes@hughesnet.org

(Note – this is the full Advanced GUI – there will be a Basic GUI that eliminates a lot of this complexity.)

Top Strip

The IRP Client automatically discovers your *nodename* and Internet *domain* name and displays them in the corresponding GUI controls.

The IRP Client enumerates all network interfaces on your device, and populates the *Interface* ComboBox with them. When you select an interface, the *Interface Description* for that interface is shown, and the *IPv6 Address* ComboBox will be loaded with all of the global (and/or ULA) IPv6 addresses assigned to that interface. The *IPv4 Private Address* is also discovered and displayed. Soon it will discover your *IPv4 Public Address* by connecting to an external node, which will reply with your public address (not yet implemented).

It then discovers and displays your device's DNS server addresses and loads them into the *DNS Server* GUI control.

Now it connects to one of your DNS servers and does a query for the *IRP over TCP* SRV record for your domain (e.g. *_irp._tcp.sixscape.com*). This returns the nodename and port of the IRP server for your domain. Finally it does a DNS resolution of that nodename to obtain the IPv6 and/or IPv4 addresses of that node. From that point on, the IRP Client can connect to the IRP server using any of the discovered addresses.

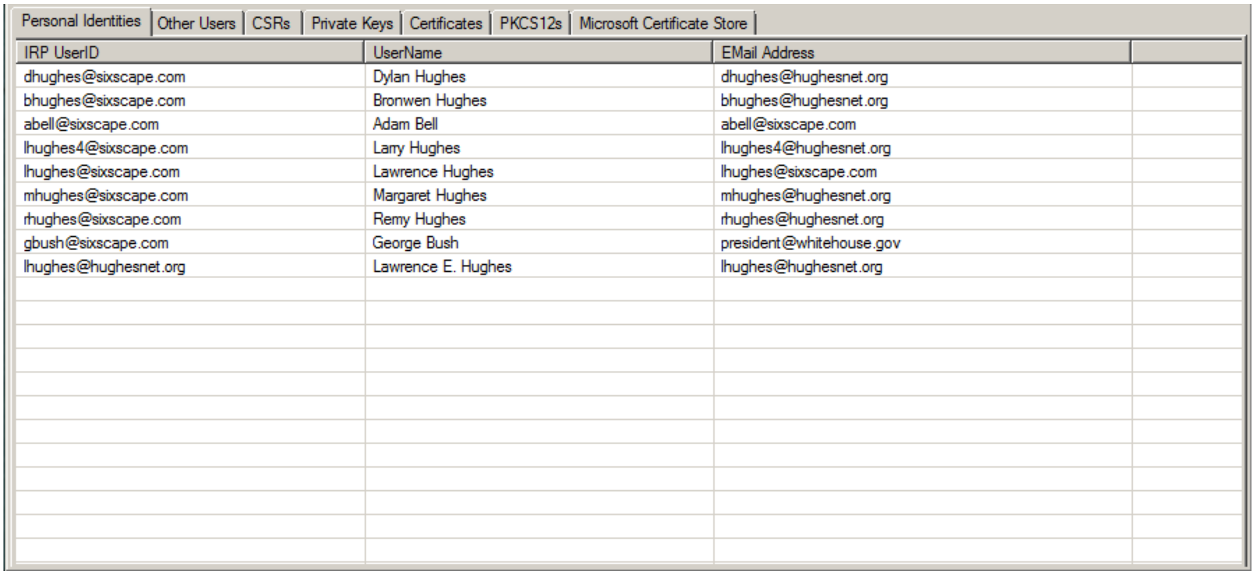
Once you login, *Current UserID* shows the UserID of the current user and *Current User Name* shows the name of the current user. *Server Status* shows the state of the server as of login, either "Online" or "Offline".

The *Logout* button terminates the application.

The *Login* button logs out the current user and allows a new login.

Note that unlike DNS, you do not need to know or configure the address of the IRP server. Any client can discover it using DNS SRV records.

Tab Control – Personal Identities Tab Page



IRP UserID	UserName	EMail Address
dhughes@sixscape.com	Dylan Hughes	dhughes@hughesnet.org
bhughes@sixscape.com	Bronwen Hughes	bhughes@hughesnet.org
abell@sixscape.com	Adam Bell	abell@sixscape.com
lhughes4@sixscape.com	Larry Hughes	lhughes4@hughesnet.org
lhughes@sixscape.com	Lawrence Hughes	lhughes@sixscape.com
mhughes@sixscape.com	Margaret Hughes	mhughes@hughesnet.org
rhughes@sixscape.com	Remy Hughes	rhughes@hughesnet.org
gbush@sixscape.com	George Bush	president@whitehouse.gov
lhughes@hughesnet.org	Lawrence E. Hughes	lhughes@hughesnet.org

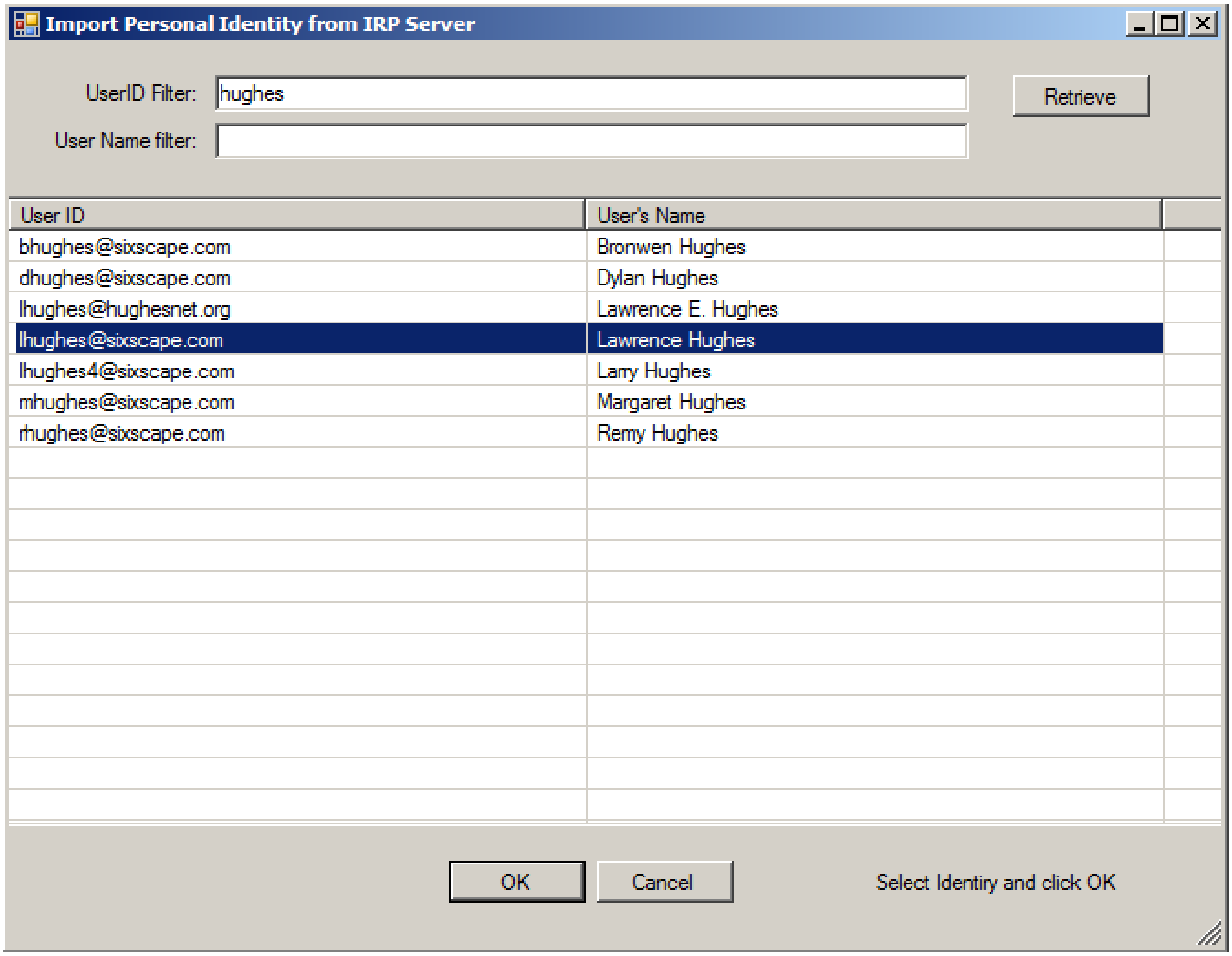
When the *Personal Identities* tab is selected, you see a list of your local Personal Identities. Each line represents one .xml file in the *IRP_Client/Personal* folder. Each line shows a UserID, a User Name and and Email Address. Double click on a line to see all of the information for that Personal Identity, and to be able to do PKI operations for it (request certificate, retrieve certificate, backup key material, etc).

If you right click, the context menu has the following items:

```
Import Identity from IRP

View Identity Info (same as double click)

Delete Identity ----- From IRP Client

                ----- From IRP Server
```

Personal Identities - Import Identity from IRP

This allows you to see a list of available Identities registered on your local IRP server. This will display a list of identities from the IRP server (showing UserID and User Name for each).



UserID Filter: hughes

User Name filter:

User ID	User's Name
bhughes@sixscape.com	Bronwen Hughes
dhughes@sixscape.com	Dylan Hughes
lhughes@hughesnet.org	Lawrence E. Hughes
lhughes@sixscape.com	Lawrence Hughes
lhughes4@sixscape.com	Larry Hughes
mhughes@sixscape.com	Margaret Hughes
rhughes@sixscape.com	Remy Hughes

Select Identiry and click OK

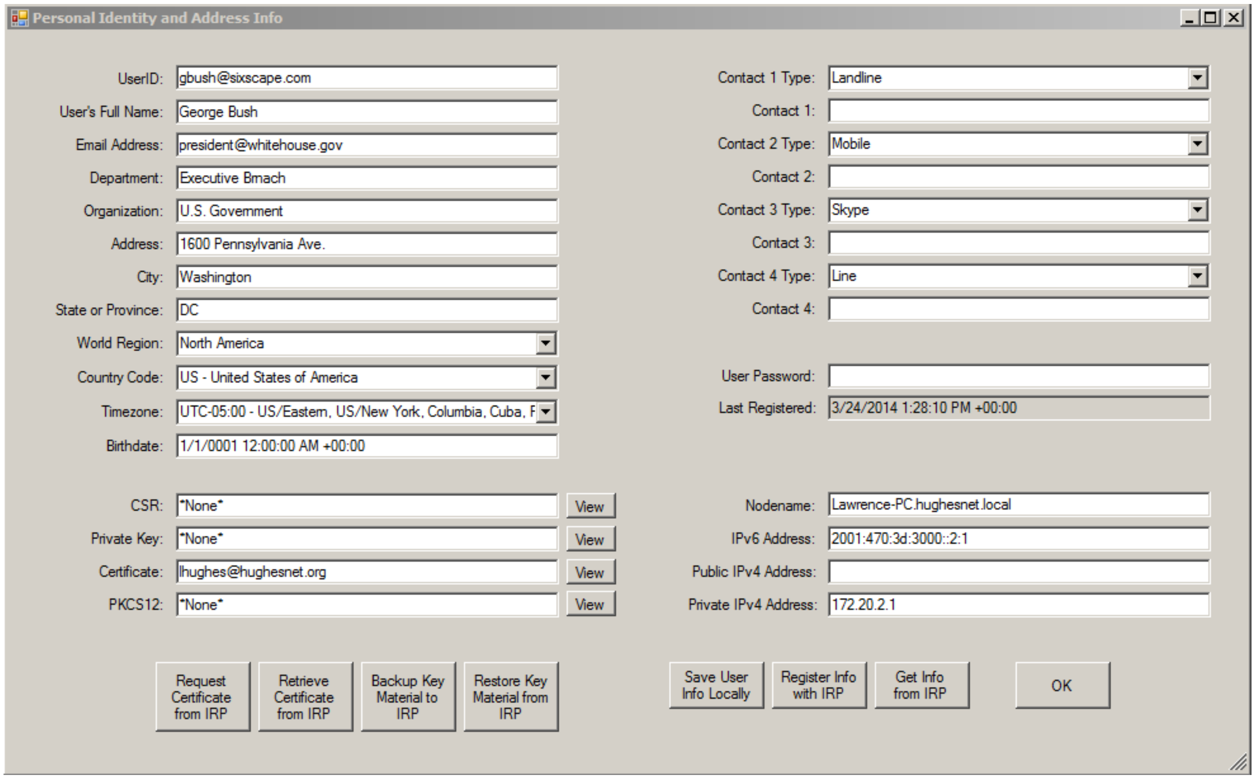
If you enter any string in either or both filter controls, and click the *Retrieve* button, only identities that contain the string in the corresponding field (as a substring) will be displayed. For example, the string "hughes" was entered in the *UserID Filter*, so only identities whose UserID contains the substring "hughes" are displayed. A blank filter means "allow all". The filter matching is case-insensitive.

To import a particular identity, click on it (which highlights it) then click *OK*. The selected identity will be downloaded and saved as an .xml file in the *IRP/Personal* folder. Since *OK* is the default control, you can also select an identity and then hit *Enter*.

Personal Identities - View Identity Info

This allows you to view the full information on a selected Personal Identity. Right click on the identity to view, then select *View Identity Info*. You can also double-click on an identity to view it. From the resulting dialog you can also do the following:

- Change any user info item(s) locally (including password)
- View the user' CSR, Private Key, Certificate or PKCS12 (if they exist)
- Request Certificate from IRP (generate a CSR for user and upload it, save local)
- Retrieve Certificate from IRP (download user's client certificate once it is approved, save local)
- Backup Key Material to IRP (create PKCS 12 package for user and upload it, save local)
- Restore Key Material from IRP (download user's PKCS 12 package and install it)
- Save User Info locally (write current info to <userId>.xml file
- Register Info with IRP (upload current user info to IRP)
- Get Info from IRP (download info from IRP, save local)


Personal Identity and Address Info

UserID: gbush@sixscape.com
User's Full Name: George Bush
Email Address: president@whitehouse.gov
Department: Executive Bmach
Organization: U.S. Govemment
Address: 1600 Pennsylvania Ave.
City: Washington
State or Province: DC
World Region: North America
Country Code: US - United States of America
Timezone: UTC-05:00 - US/Eastem, US/New York, Columbia, Cuba, F
Birthdate: 1/1/0001 12:00:00 AM +00:00

CSR: *None*
Private Key: *None*
Certificate: lhughes@hughesnet.org
PKCS12: *None*

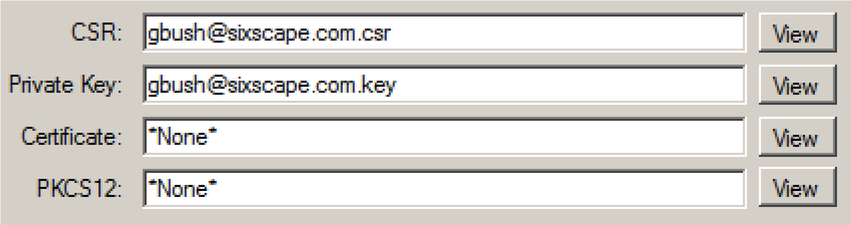
Contact 1 Type: Landline
Contact 1:
Contact 2 Type: Mobile
Contact 2:
Contact 3 Type: Skype
Contact 3:
Contact 4 Type: Line
Contact 4:

User Password:
Last Registered: 3/24/2014 1:28:10 PM +00:00

Nodename: Lawrence-PC.hughesnet.local
IPv6 Address: 2001:470:3d:3000::2:1
Public IPv4 Address:
Private IPv4 Address: 172.20.2.1

View
View
View
View
Request Certificate from IRP
Retrieve Certificate from IRP
Backup Key Material to IRP
Restore Key Material from IRP
Save User Info Locally
Register Info with IRP
Get Info from IRP
OK

When done with this screen, dismiss it with *OK*.



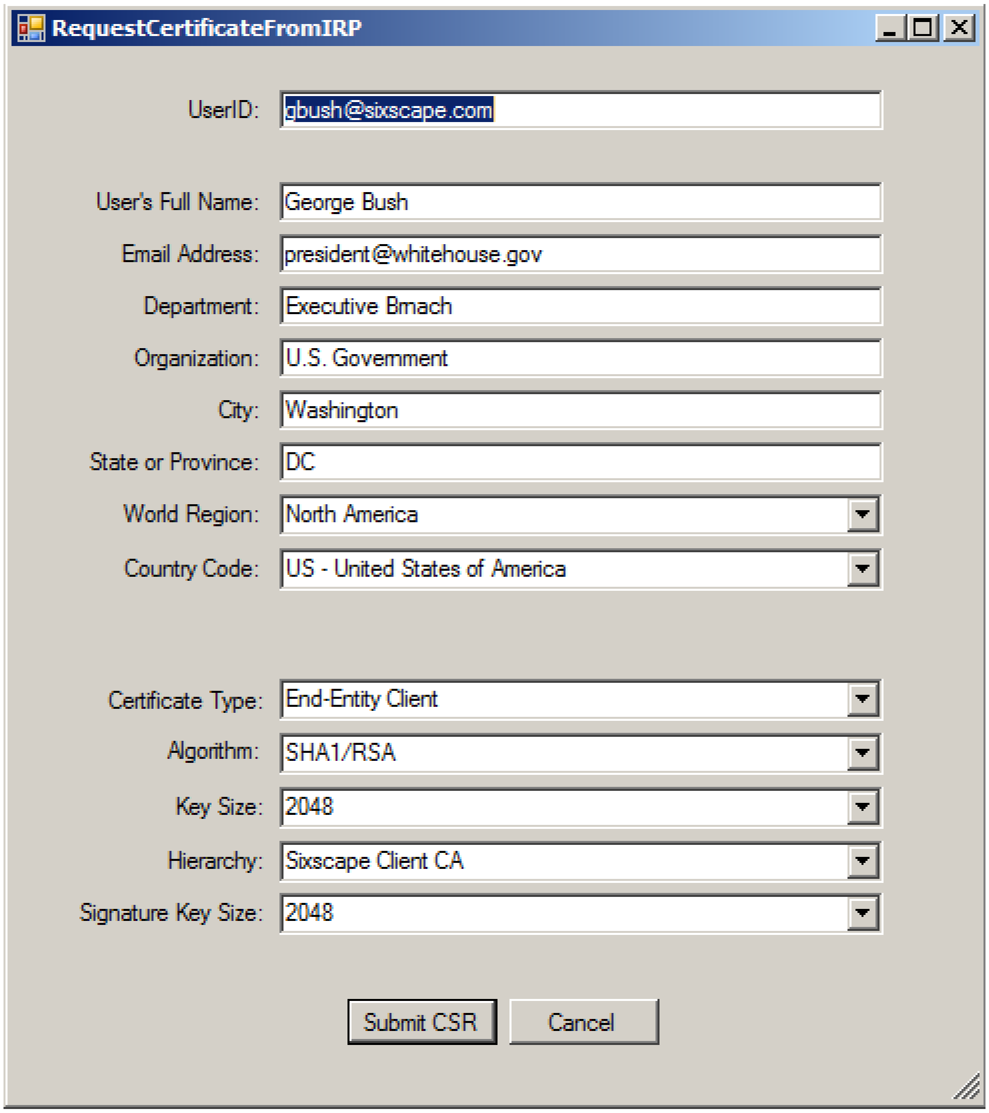
CSR: gbush@sixscape.com.csr

Private Key: gbush@sixscape.com.key

Certificate: *None*

PKCS12: *None*

When you click this button, the following dialog will appear:



UserID: gbush@sixscape.com

User's Full Name: George Bush

Email Address: president@whitehouse.gov

Department: Executive Brnach

Organization: U.S. Government

City: Washington

State or Province: DC

World Region: North America

Country Code: US - United States of America

Certificate Type: End-Entity Client

Algorithm: SHA1/RSA

Key Size: 2048

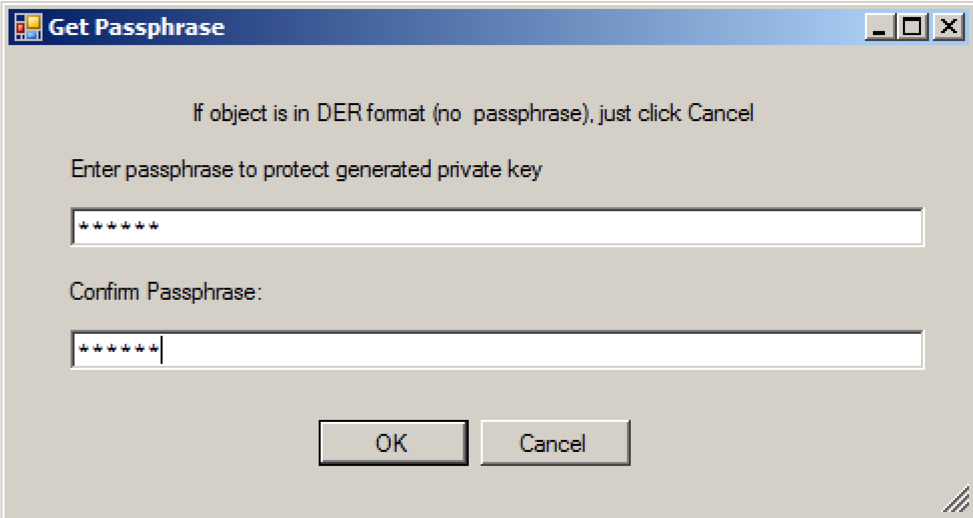
Hierarchy: Sixscape Client CA

Signature Key Size: 2048

The fields are filled in for you automatically from the current user info, or reasonable defaults. Make any changes you need (e.g. key size) and click Submit CSR (or just hit *Enter*). Note – making changes to the info here will not update the main Personal Identity Info page – it just affects what goes into the CSR.

When you submit the CSR, there will be a brief pause while the IRP client generates a keypair (the Submit CSR button will go gray during this time). Your CSR will be stored in the *IRP_Client/PKI_DB/CSRs* folder.

You will then be asked to enter a passphrase to protect the generated private key:



Get Passphrase

If object is in DER format (no passphrase), just click Cancel

Enter passphrase to protect generated private key

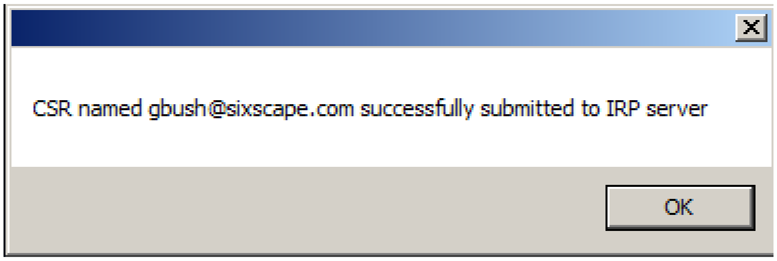
******

Confirm Passphrase:

******

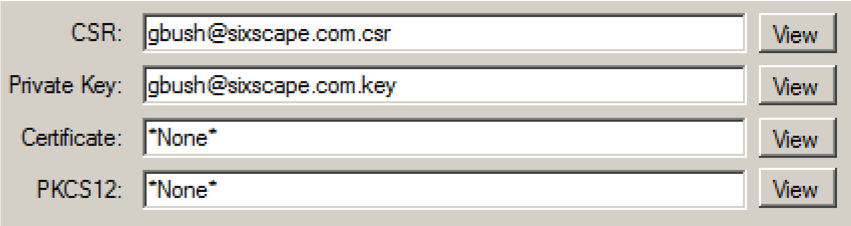
Enter your passphrase and confirm it, then click *OK*. Your password will be stored in the *IRP_Client/PKI_DB/PrivateKeys* folder (encrypted).

It will then show the following to confirm the CSR was successfully submitted. If there was already a CSR on the IRP server with this name, it will inform you of that, and allow you to replace the old one with the new one.



CSR named gbush@sixscape.com successfully submitted to IRP server

Once it is submitted, the user's CSR and Private Key items will be updated. You can view either by clicking the corresponding *View* button.



CSR: gbush@sixscape.com.csr

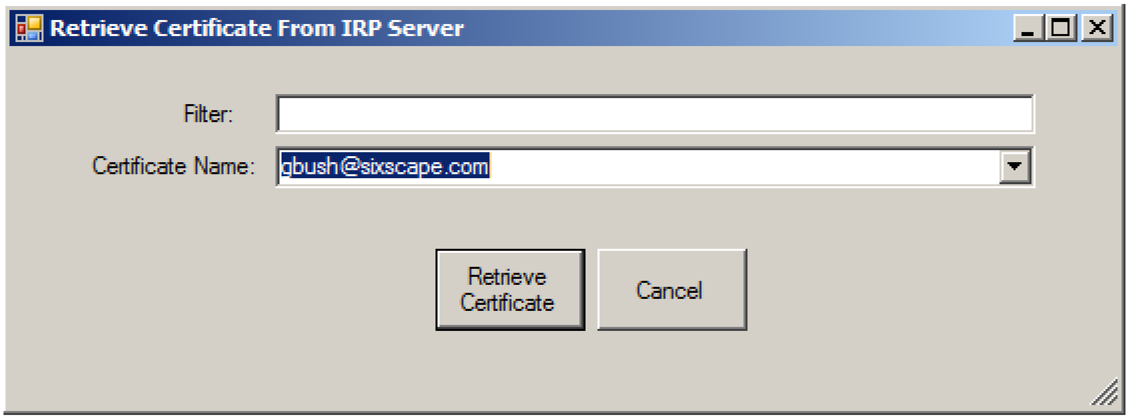
Private Key: gbush@sixscape.com.key

Certificate: *None*

PKCS12: *None*

Personal Identities - View Personal Identity Info - Retrieve Certificate from IRP

Once the IRP admin has approved your CSR and generated your certificate, you can retrieve it with this button. You will see the following dialog:



Filter:

Certificate Name: gbush@sixscape.com

If the certificate has been generated, it will appear as the selected certificate name. If some other UserID appears (or no name appears) then your certificate has not yet been generated (wait for notification from your IRP admin – usually by email).

To retrieve the certificate, click the *Retreive Certificate* button.

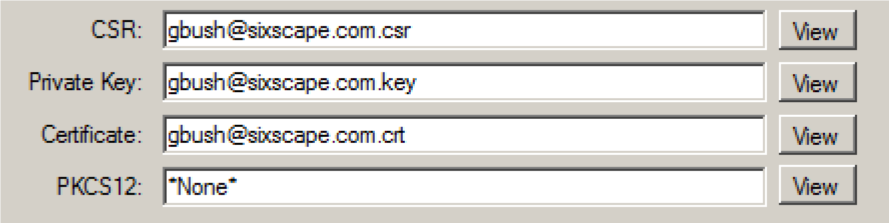
Your certificate will be downloaded from the IRP Server and stored in the *IRP_Client/PKI_DB/Certificates* folder.

It will ask for the passphrase that protects your private key, so it can verify this is the correct certificate. Enter your private key passphrase and confirm it, the click *OK*.

It will ask for the passphrase to access your private key again, in order to store your key and certificate in the Microsoft Certificate Store. Enter your private key passphrase, confirm it, and click *OK*.

[Coming soon: it will remember the private key passphrase and not require you to enter it twice]

Once this is done, the user's Certificate item will be updated. You can view it by clicking the corresponding *View* button.



CSR: gbush@sixscape.com.csr

Private Key: gbush@sixscape.com.key

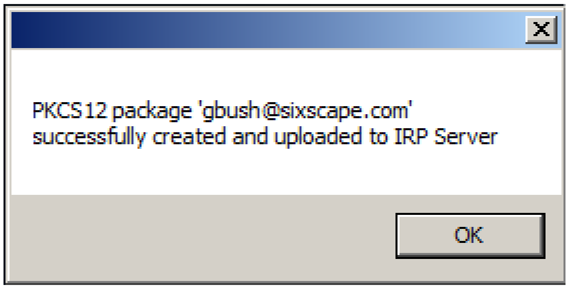
Certificate: gbush@sixscape.com.crt

PKCS12: *None*

Personal Identities - View Personal Identity Info - Backup Key Material to IRP

This allows you to create a PKCS12 package containing both your certificate and private key and encrypt it with a passphrase known only to you. This is stored locally (in the *IRP_Client/PKI_DB/PKCS12s* folder. It is also uploaded to IRP, to allow you to restore your key material later, to this (or any other) computer.

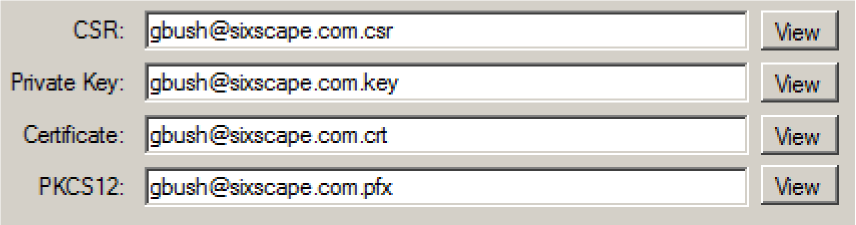
Click the *Backup Key Material to IRP* button. You will be asked to enter the passphrase for your private key, and confirm it. Do so. You will then be asked to enter a passphrase to protect your new PKCS12 file. Do do. You will then see the following:



PKCS12 package 'gbush@sixscape.com'
successfully created and uploaded to IRP Server

Click *OK* to proceed.

Once this is done, the user's PKCS12 item will be updated. You can view it by clicking the corresponding *View* button:



CSR:	gbush@sixscape.com.csr
Private Key:	gbush@sixscape.com.key
Certificate:	gbush@sixscape.com.crt
PKCS12:	gbush@sixscape.com.pfx

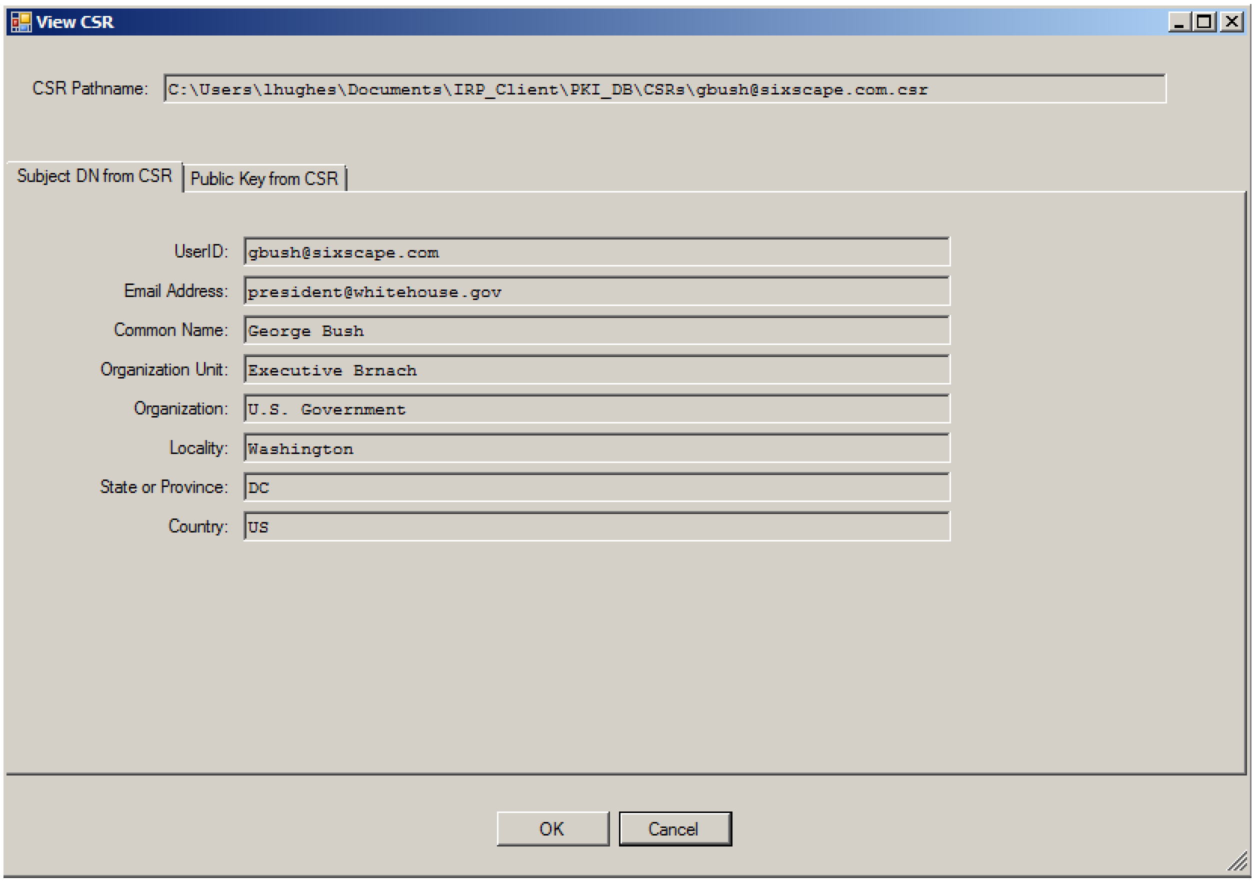
At each one of these steps, the User Info (including PKI object names) has been updated locally (to the user's .xml file), so there is no need to *Save User Info Locally* now. In the normal course of this, the user's CSR, Certificate and PKCS12 file have already been saved locally and uploaded to the IRP server (no need to upload them again). Your private key is not normally uploaded to the *IRP Server*, except inside the encrypted PKCS12 file. The IRP admin normally does not know the passphrase you protected your PKCS12 file with, hence they cannot obtain the private key in it – you on the other hand can download that file and install the cert and private key within to the same or any other computer. Of course you need to know that passphrase. You may want to record that passphrase, but so long as you have your key material installed on at least one machine you can always create a new PKCS12 package and upload it again to the *IRP Server*, with a new passphrase.

Personal Identities - Save User Info Locally / Register Info with IRP / Get Info from IRP

These buttons do exactly what they say, and there is no dialog box required. Note that the password is stored locally only in salted, hashed form. It is extraordinarily difficult to recover the login password from this, so it is safe to keep in your .xml file. Doing this makes it possible to authenticate locally even if the *IRP server* is currently unavailable to you.

The *OK* button dismisses the *Personal Identity and Address Info* dialog – it does not cause any information to be written or uploaded. That must be done by using the appropriate buttons.

When you View a **CSR**, you will see the following dialog:

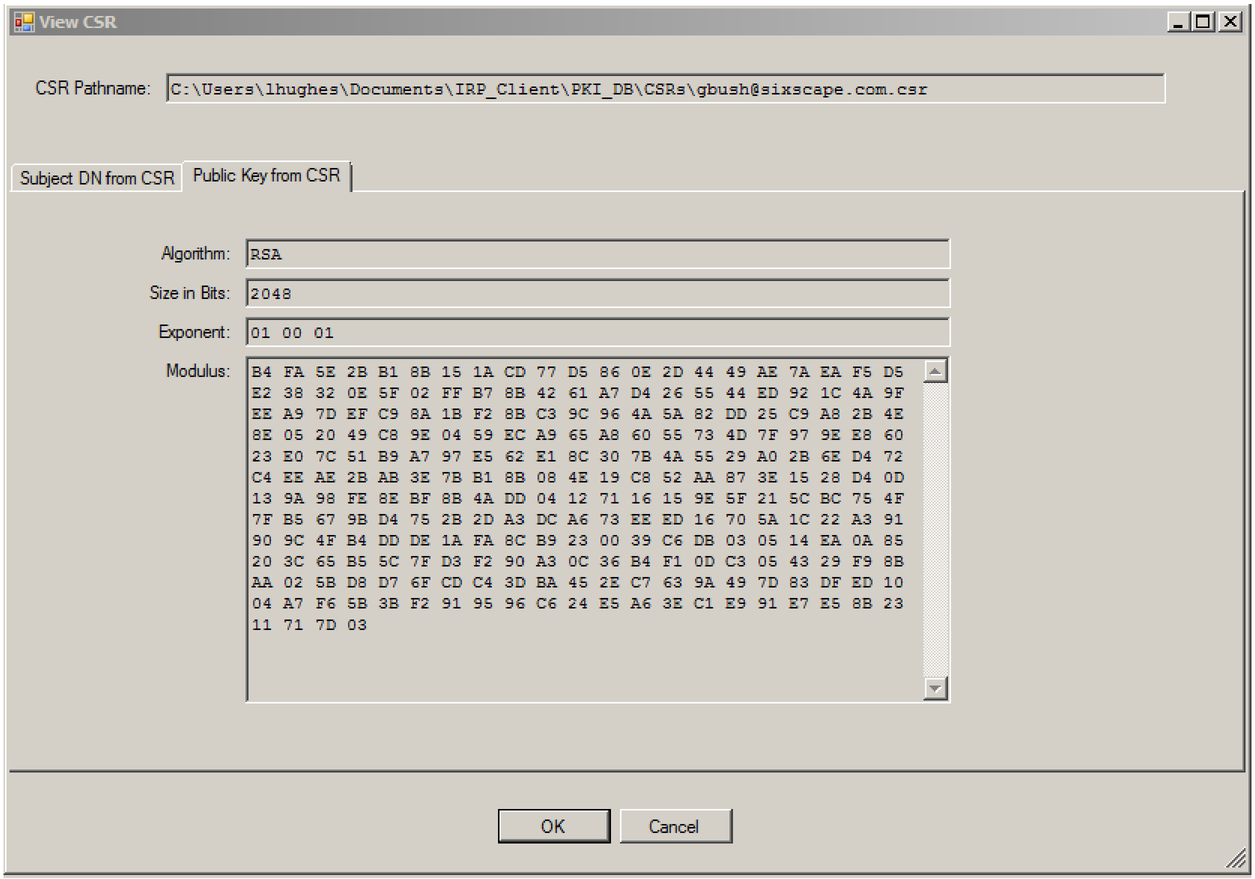


CSR Pathname: C:\Users\lhughes\Documents\IRP_Client\PKI_DB\CSRs\gbush@sixscape.com.csr

UserID: gbush@sixscape.com
Email Address: president@whitehouse.gov
Common Name: George Bush
Organization Unit: Executive Brnach
Organization: U.S. Government
Locality: Washington
State or Province: DC
Country: US

The *Subject DN from CSR* shows the items from the Subject Distinguished Name of the Certificate Signing Request. You cannot change any of these items, so all GUI controls are Read Only.

The *Public Key from CSR* tab shows the public key information from the CSR. There is no Serial Number, Valid From, Valid To, Issuer Distinguished Name or private key associated with a CSR.



CSR Pathname: C:\Users\lhughes\Documents\IRP_Client\PKI_DB\CSRs\gbush@sixscape.com.csr

Algorithm: RSA

Size in Bits: 2048

Exponent: 01 00 01

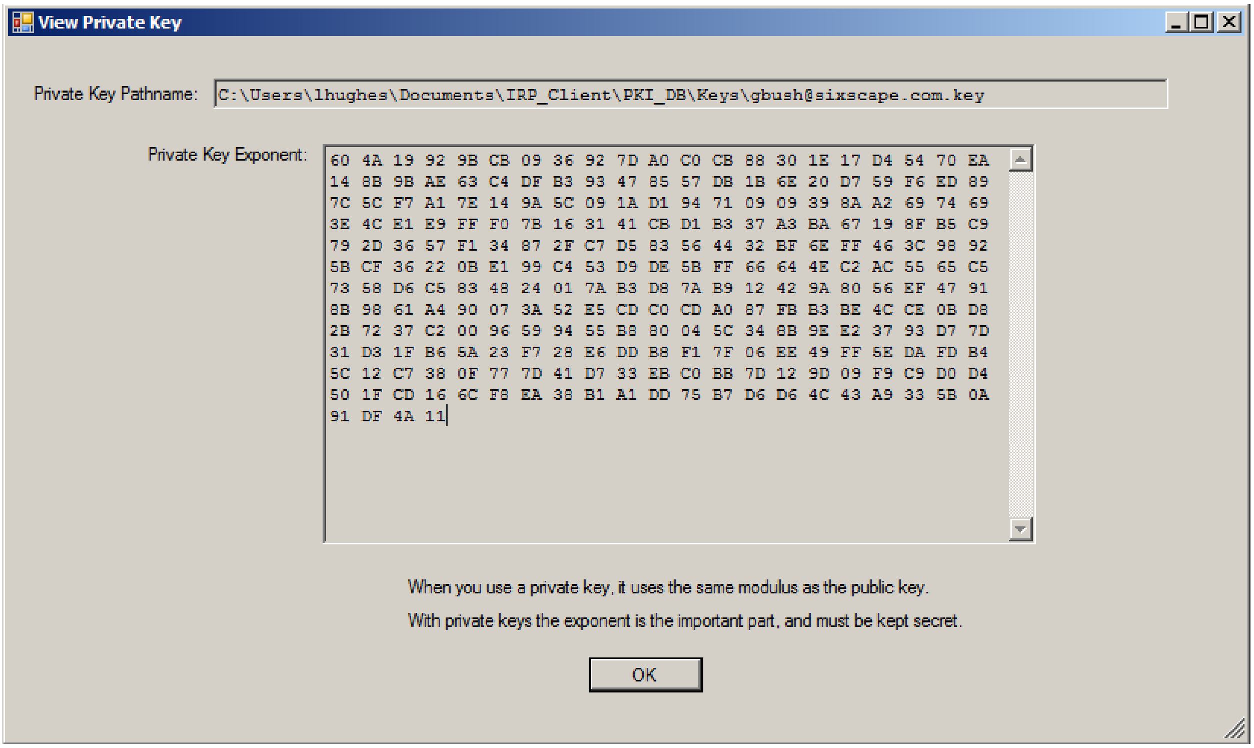
Modulus:

```
B4 FA 5E 2B B1 8B 15 1A CD 77 D5 86 0E 2D 44 49 AE 7A EA F5 D5
E2 38 32 0E 5F 02 FF B7 8B 42 61 A7 D4 26 55 44 ED 92 1C 4A 9F
EE A9 7D EF C9 8A 1B F2 8B C3 9C 96 4A 5A 82 DD 25 C9 A8 2B 4E
8E 05 20 49 C8 9E 04 59 EC A9 65 A8 60 55 73 4D 7F 97 9E E8 60
23 E0 7C 51 B9 A7 97 E5 62 E1 8C 30 7B 4A 55 29 A0 2B 6E D4 72
C4 EE AE 2B AB 3E 7B B1 8B 08 4E 19 C8 52 AA 87 3E 15 28 D4 0D
13 9A 98 FE 8E BF 8B 4A DD 04 12 71 16 15 9E 5F 21 5C BC 75 4F
7F B5 67 9B D4 75 2B 2D A3 DC A6 73 EE ED 16 70 5A 1C 22 A3 91
90 9C 4F B4 DD DE 1A FA 8C B9 23 00 39 C6 DB 03 05 14 EA 0A 85
20 3C 65 B5 5C 7F D3 F2 90 A3 0C 36 B4 F1 0D C3 05 43 29 F9 8B
AA 02 5B D8 D7 6F CD C4 3D BA 45 2E C7 63 9A 49 7D 83 DF ED 10
04 A7 F6 5B 3B F2 91 95 96 C6 24 E5 A6 3E C1 E9 91 E7 E5 8B 23
11 71 7D 03
```

You would not normally need to see your public key. It is provided just for the curious.

Viewing a Private Key

You must first enter (and confirm) the passphrase used to protect that private key. You will then see the following dialog:



Private Key Pathname: C:\Users\lhughes\Documents\IRP_Client\PKI_DB\Keys\gbush@sixscape.com.key

Private Key Exponent:

```
60 4A 19 92 9B CB 09 36 92 7D A0 C0 CB 88 30 1E 17 D4 54 70 EA
14 8B 9B AE 63 C4 DF B3 93 47 85 57 DB 1B 6E 20 D7 59 F6 ED 89
7C 5C F7 A1 7E 14 9A 5C 09 1A D1 94 71 09 09 39 8A A2 69 74 69
3E 4C E1 E9 FF F0 7B 16 31 41 CB D1 B3 37 A3 BA 67 19 8F B5 C9
79 2D 36 57 F1 34 87 2F C7 D5 83 56 44 32 BF 6E FF 46 3C 98 92
5B CF 36 22 0B E1 99 C4 53 D9 DE 5B FF 66 64 4E C2 AC 55 65 C5
73 58 D6 C5 83 48 24 01 7A B3 D8 7A B9 12 42 9A 80 56 EF 47 91
8B 98 61 A4 90 07 3A 52 E5 CD C0 CD A0 87 FB B3 BE 4C CE 0B D8
2B 72 37 C2 00 96 59 94 55 B8 80 04 5C 34 8B 9E E2 37 93 D7 7D
31 D3 1F B6 5A 23 F7 28 E6 DD B8 F1 7F 06 EE 49 FF 5E DA FD B4
5C 12 C7 38 0F 77 7D 41 D7 33 EB C0 BB 7D 12 9D 09 F9 C9 D0 D4
50 1F CD 16 6C F8 EA 38 B1 A1 DD 75 B7 D6 D6 4C 43 A9 33 5B 0A
91 DF 4A 11
```

When you use a private key, it uses the same modulus as the public key.

With private keys the exponent is the important part, and must be kept secret.

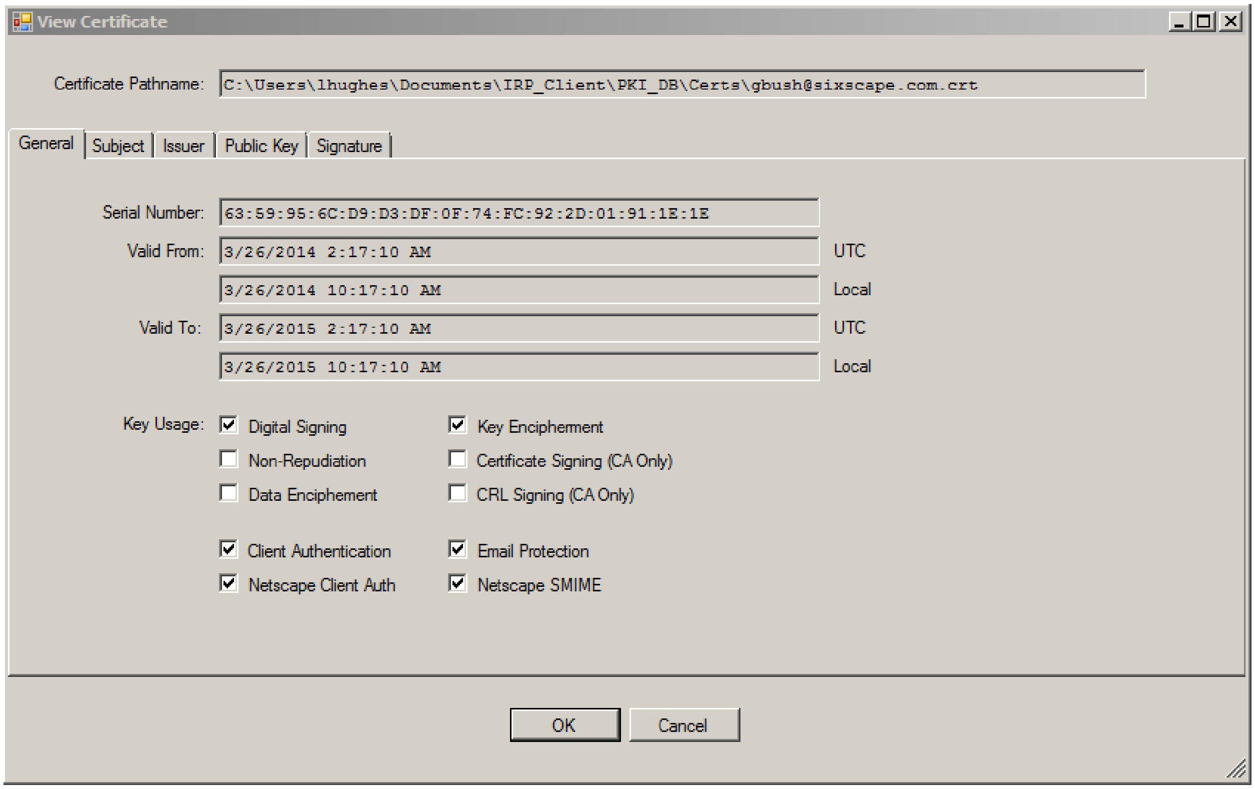
Since your private key uses the same modulus as your public key, there is no need to display it again here. While the public key has a short exponent (usually 65,537, or 0x100001), with private keys, the exponent is the important part. Each private key will have a different exponent, and that exponent will be the same length (in bits) as the modulus (e.g. 2048 bits). When done viewing, dismiss the dialog with the *OK* button.

It is really one giant integer number. Here it is represented as a series of two hex digit (8 bit) fields. A 2048 bit key can be represented by 256 (= 2048 / 8) such fields.

You should not let unauthorized parties view this screen, as this compromises the keypair (do not let others look over your shoulder, or take pictures of this screen).

Viewing a certificate – General Page

When you view a **Certificate**, you will see the following dialog (**Certificate – General page**)



Certificate Pathname: C:\Users\lhughes\Documents\IRP_Client\PKI_DB\Certs\gbush@sixscape.com.crt

Serial Number: 63:59:95:6C:D9:D3:DF:0F:74:FC:92:2D:01:91:1E:1E

Valid From: 3/26/2014 2:17:10 AM UTC

3/26/2014 10:17:10 AM Local

Valid To: 3/26/2015 2:17:10 AM UTC

3/26/2015 10:17:10 AM Local

Key Usage: ☑ Digital Signing ☑ Key Enciphement

□ Non-Repudiation □ Certificate Signing (CA Only)

□ Data Enciphement □ CRL Signing (CA Only)

☑ Client Authentication ☑ Email Protection

☑ Netscape Client Auth ☑ Netscape SMIME

The certificate has more information than the CSR, so there are additional pages in the tab control.

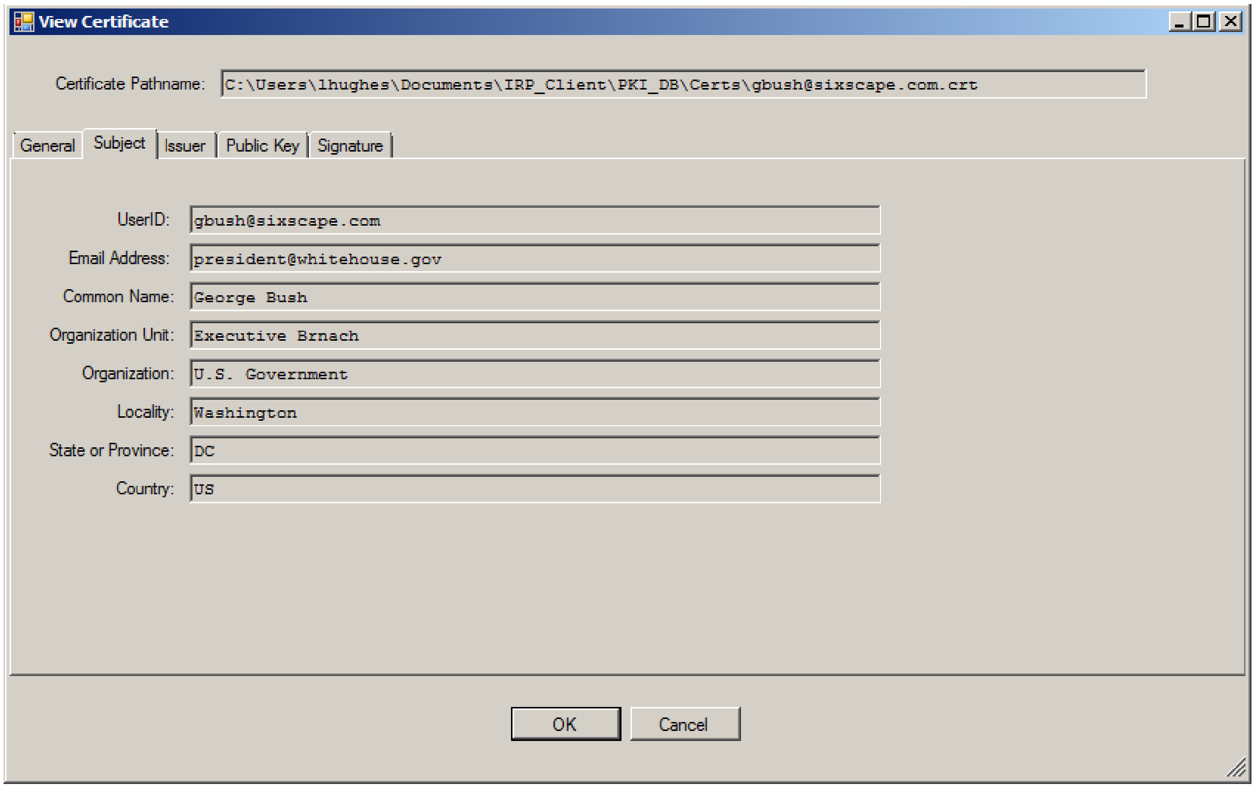
The *General* tab shows information supplied by the CA at the time it was signed, including:

- *Serial Number* (a globally unique 128-bit random number, or possibly a small integer for CA certs).
- *Valid From* (the date before which the certificate is not valid), shown in both UTC and local.
- *Valid To* (the date after which the certificate is not valid), show in both UTC and local.
- *Key Usage – basic options*
    - *Digital Signing* means the cert (actually the corresponding private key) can be used to create a digital signature (e.g. by the sender of a Signed message), and the cert can be used to verify digital signatures (e.g. by the recipient of a Signed message)
    - *Key Encipherment* means the cert can be used to encrypt a session key (e.g. by the sender of an Encrypted message), and the corresponding private key can be used to decrypt a session key (e.g. by the recipient of an Encrypted message).
    - *Non-repudiation* means the cert (actually the corresponding private key) can be used to establish that the sender is the only person who could have sent this signed message.
    - *Certificate Signing* means the cert (actually the corresponding private key) can be used to sign other certs (this would be set only on CA certs, like root and intermediate).

    - *Data Encipherment* means the cert can be used to encrypt data (as opposed to keys), and the corresponding private key can be used to decrypt the encrypted data.
    - *CRL Signing* means the cert (actually the corresponding private key) can be used to digitally sign a Certificate Revocation List, and the cert can be used to verify the digital signature on a CRL (this would be set only on CA certs, like root and intermediate).
- *Key Usage – Extended Key Usage options*
    - *Client Authentication* means the cert can be used to authenticate a client to a server in SSL/TLS handshake. The corresponding private key would also be used (in the cryptographic challenge).
    - *Email Protection* means the cert can be used in S/MIME secure email. The corresponding private key can be used to create digital signatures and decrypt the symmetric session key in encrypted messages. The cert can be used to verify digital signatures or encrypt the symmetric session key in encrypted messages.
- *Key Usage – Netscape specific Usage options*
    - *Netscape Client Auth – same as Client Authentication, but used by Netscape products*
    - *Netscape SMIME – same as Email Protection, but used by Netscape products*

Viewing a Certificate – Subject Page

This page shows the fields from the certificate’s Subject Distinguished Name.



Certificate Pathname: C:\Users\lhughes\Documents\IRP_Client\PKI_DB\Certs\gbush@sixscape.com.crt

UserID: gbush@sixscape.com

Email Address: president@whitehouse.gov

Common Name: George Bush

Organization Unit: Executive Brnach

Organization: U.S. Government

Locality: Washington

State or Province: DC

Country: US

The **UserID** *is the IRP UserID, e.g. gbush*@sixscape.com.

The **Email Address** is the user’s email address, e.g. *president@whitehouse.gov*

The **Common Name** is the person’s name, e.g. *George Bush*.

The **Organization Unit** is the name of the user’s department within their organization, e.g. *Executive Branch*

The **Organization** is the name of the user’s company or agency, e.g. *U.S. Government*

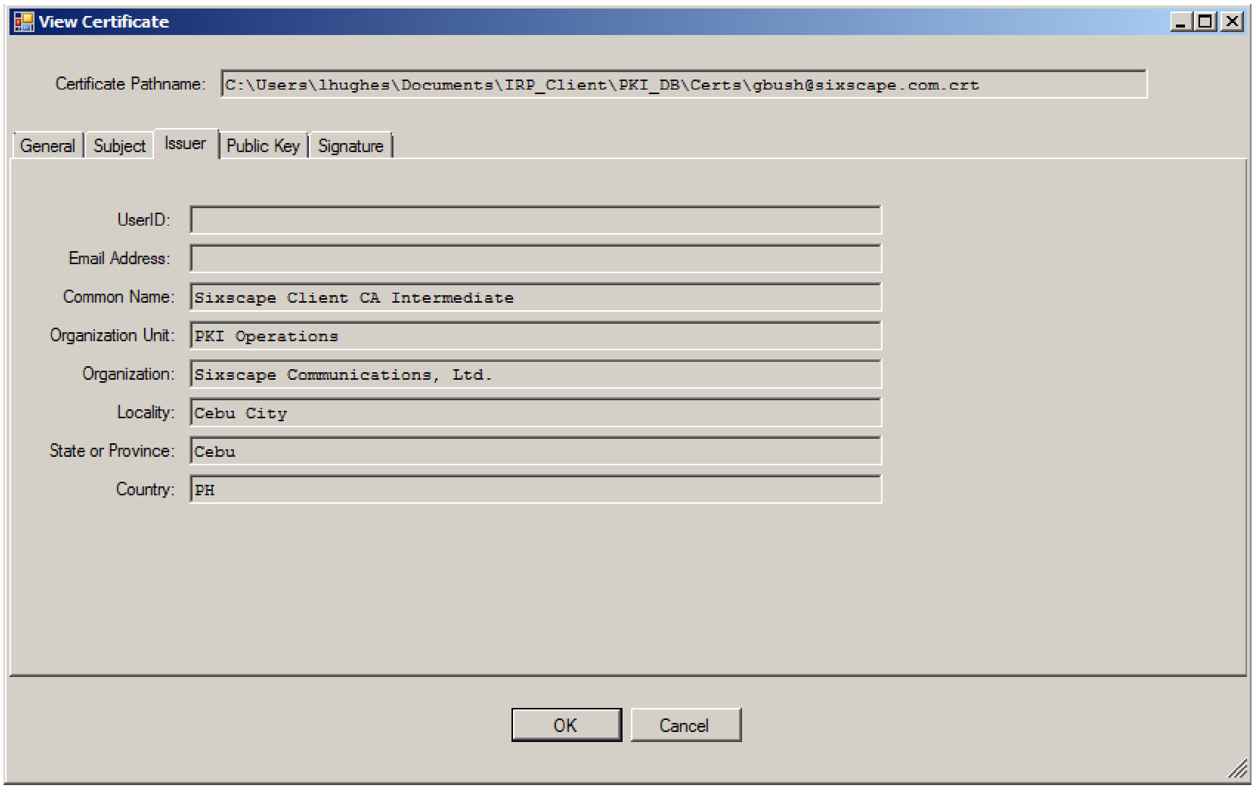
The **Locality** is the name of the user’s city, village, or other locality, e.g. *Washington*

The **State or Province** is the sub-national level division, such as state, province or district, e.g. *D.C.*

The **Country** is the two-letter code for the user’s country, e.g. *US*

Viewing a Certificate – Issuer Page

This page shows the fields from the certificate’s Issuer Distinguished Name.



Certificate Pathname: C:\Users\lhughes\Documents\IRP_Client\PKI_DB\Certs\gbush@sixscape.com.crt

UserID:

Email Address:

Common Name: Sixscape Client CA Intermediate

Organization Unit: PKI Operations

Organization: Sixscape Communications, Ltd.

Locality: Cebu City

State or Province: Cebu

Country: PH

The **UserID** *is the IRP UserID* from the signing cert (usually blank)

The **Email Address** is the email address from the signing cert (usually blank)

The **Common Name** is usually the name of the signing cert (including hierarchy name), e.g. *Sixscape Client CA Intermediate*

The **Organization Unit** is the name of the CA’s department within organization, e.g. *PKI Operations*

The **Organization** is the name of the CA’s company or agency, e.g. *Sixscape Communications, Ltd.*

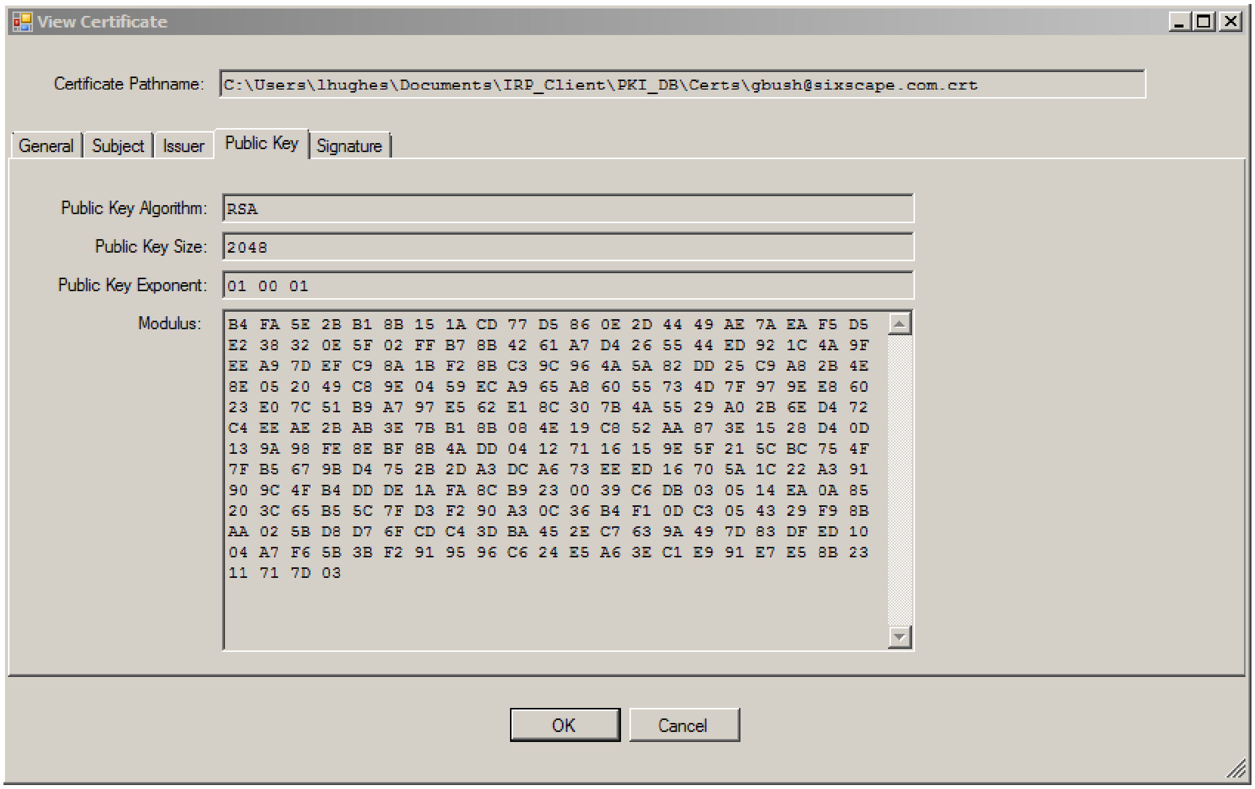
The **Locality** is the name of the CA’s city, village, or other locality, e.g. *Cebu City*

The **State or Province** is the CA’s sub-national level division, such as state, province or district, e.g. *Cebu*

The **Country** is the two-letter code for the CA’s country, e.g. *PH*

Viewing a Certificate – Public Key page

This page shows the information about the public key in the certificate



Certificate Pathname: C:\Users\lhughes\Documents\IRP_Client\PKI_DB\Certs\gbush@sixscape.com.crt

Public Key Algorithm: RSA

Public Key Size: 2048

Public Key Exponent: 01 00 01

Modulus:

```
B4 FA 5E 2B B1 8B 15 1A CD 77 D5 86 0E 2D 44 49 AE 7A EA F5 D5
E2 38 32 0E 5F 02 FF B7 8B 42 61 A7 D4 26 55 44 ED 92 1C 4A 9F
EE A9 7D EF C9 8A 1B F2 8B C3 9C 96 4A 5A 82 DD 25 C9 A8 2B 4E
8E 05 20 49 C8 9E 04 59 EC A9 65 A8 60 55 73 4D 7F 97 9E E8 60
23 E0 7C 51 B9 A7 97 E5 62 E1 8C 30 7B 4A 55 29 A0 2B 6E D4 72
C4 EE AE 2B AB 3E 7B B1 8B 08 4E 19 C8 52 AA 87 3E 15 28 D4 0D
13 9A 98 FE 8E BF 8B 4A DD 04 12 71 16 15 9E 5F 21 5C BC 75 4F
7F B5 67 9B D4 75 2B 2D A3 DC A6 73 EE ED 16 70 5A 1C 22 A3 91
90 9C 4F B4 DD DE 1A FA 8C B9 23 00 39 C6 DB 03 05 14 EA 0A 85
20 3C 65 B5 5C 7F D3 F2 90 A3 0C 36 B4 F1 0D C3 05 43 29 F9 8B
AA 02 5B D8 D7 6F CD C4 3D BA 45 2E C7 63 9A 49 7D 83 DF ED 10
04 A7 F6 5B 3B F2 91 95 96 C6 24 E5 A6 3E C1 E9 91 E7 E5 8B 23
11 71 7D 03
```

The **Public Key Algorithm** is the asymmetric key algorithm this key is used with, usually RSA.

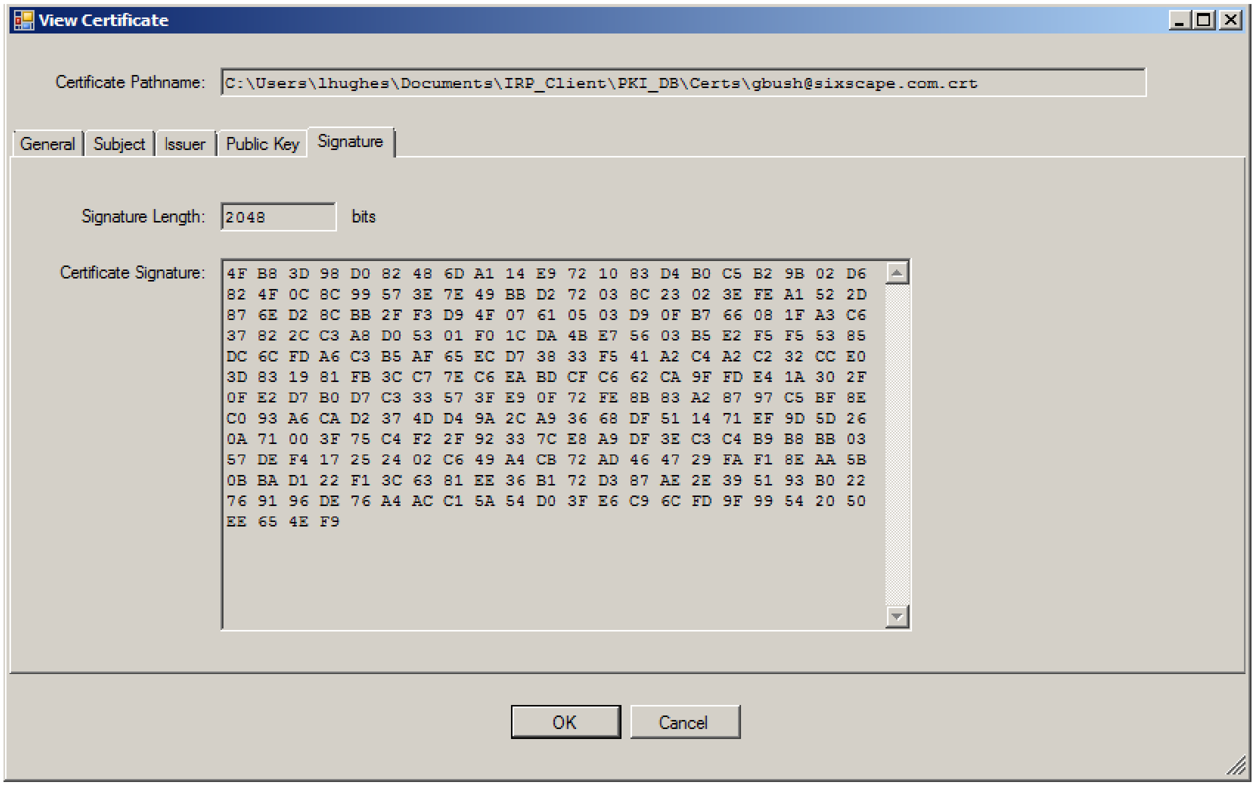
The **Public Key Size** is the size of the public key, in bits, typically 1024, 2048, 3072 or 4096

The **Public Key Exponent** is the exponent for the public key. Usually a small, common used value, such as 65,537 decimal (0x10001 hex).

The **Modulus** is the modulus for this keypair (used by both the public key and private key). This is unique for each keypair, but it does not compromise security for this to be published. It is really one giant integer number (which is the product of two smaller prime numbers). Here it is represented as a series of two hex digit (8 bit) fields. A 2048 bit key can be represented by 256 (= 2048 / 8) such fields.

Viewing a Certificate – Signature Page

This page shows the digital signature of the certificate.



Certificate Pathname: C:\Users\lhughes\Documents\IRP_Client\PKI_DB\Certs\gbush@sixscape.com.crt

Signature Length: 2048 bits

Certificate Signature:

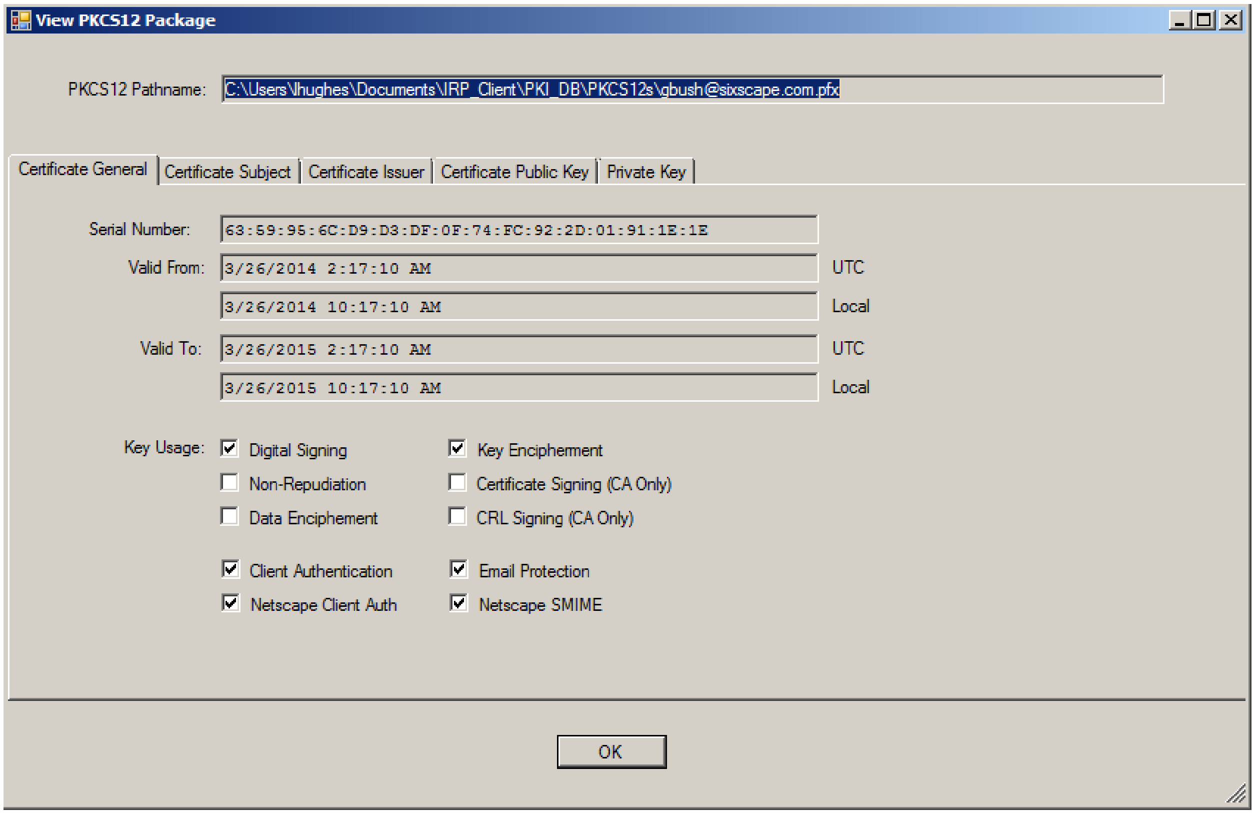
```
4F B8 3D 98 D0 82 48 6D A1 14 E9 72 10 83 D4 B0 C5 B2 9B 02 D6
82 4F 0C 8C 99 57 3E 7E 49 BB D2 72 03 8C 23 02 3E FE A1 52 2D
87 6E D2 8C BB 2F F3 D9 4F 07 61 05 03 D9 0F B7 66 08 1F A3 C6
37 82 2C C3 A8 D0 53 01 F0 1C DA 4B E7 56 03 B5 E2 F5 F5 53 85
DC 6C FD A6 C3 B5 AF 65 EC D7 38 33 F5 41 A2 C4 A2 C2 32 CC E0
3D 83 19 81 FB 3C C7 7E C6 EA BD CF C6 62 CA 9F FD E4 1A 30 2F
0F E2 D7 B0 D7 C3 33 57 3F E9 0F 72 FE 8B 83 A2 87 97 C5 BF 8E
C0 93 A6 CA D2 37 4D D4 9A 2C A9 36 68 DF 51 14 71 EF 9D 5D 26
0A 71 00 3F 75 C4 F2 2F 92 33 7C E8 A9 DF 3E C3 C4 B9 B8 BB 03
57 DE F4 17 25 24 02 C6 49 A4 CB 72 AD 46 47 29 FA F1 8E AA 5B
0B BA D1 22 F1 3C 63 81 EE 36 B1 72 D3 87 AE 2E 39 51 93 B0 22
76 91 96 DE 76 A4 AC C1 5A 54 D0 3F E6 C9 6C FD 9F 99 54 20 50
EE 65 4E F9
```

The **Signature Length** is the size of the signature in bits (e.g. 2048)

The **Certificate Signature** is the actual digital signature (which is the message digest of the certificate, encrypted by an asymmetric key algorithm and the private key of the signing certificate).

Viewing a PKCS12 Object

When you view a PKCS12 object, it requires you to enter the passphrase used to protect it. It then combines the pages for both the certificate and the private key (see above for details):



PKCS12 Pathname: C:\Users\lhughes\Documents\IRP_Client\PKI_DB\PKCS12s\gbush@sixscape.com.pfx

Serial Number: 63:59:95:6C:D9:D3:DF:0F:74:FC:92:2D:01:91:1E:1E

Valid From: 3/26/2014 2:17:10 AM UTC
3/26/2014 10:17:10 AM Local

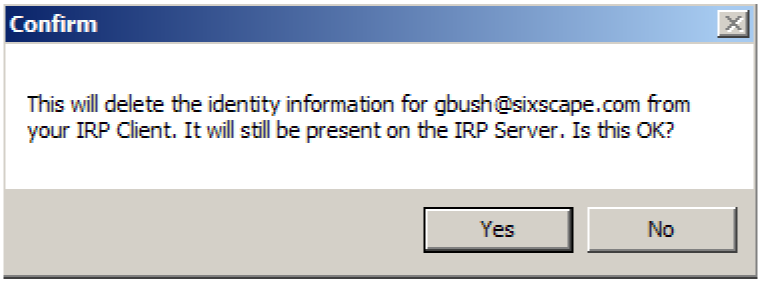
Valid To: 3/26/2015 2:17:10 AM UTC
3/26/2015 10:17:10 AM Local

Key Usage:
☑ Digital Signing ☑ Key Encipherment
☐ Non-Repudiation ☐ Certificate Signing (CA Only)
☐ Data Encipherment ☐ CRL Signing (CA Only)

☑ Client Authentication ☑ Email Protection
☑ Netscape Client Auth ☑ Netscape SMIME

Personal Identities - Delete Identity – From IRP Client

If you right click on a Personal Identity and select *Delete Identity – From IRP Client*, it will ask you to confirm this action:

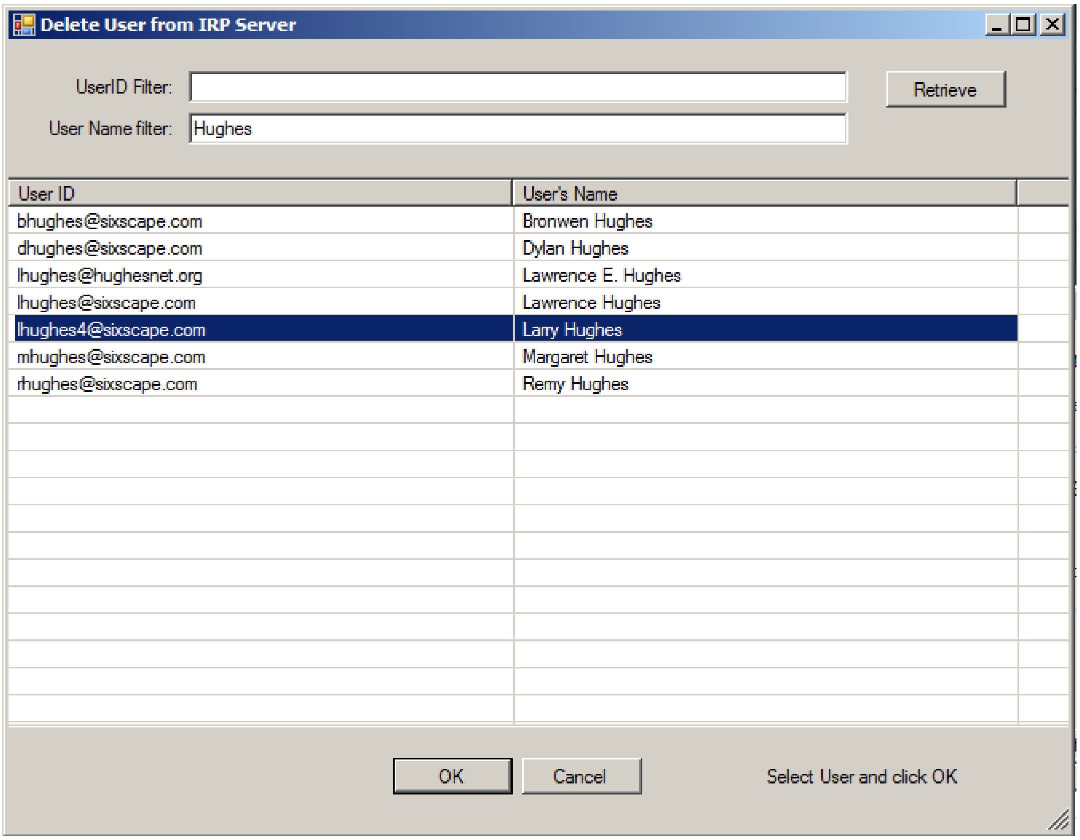


This will delete the identity information for gbush@sixscape.com from your IRP Client. It will still be present on the IRP Server. Is this OK?

If you click *Yes*, the .xml file for this identity will be deleted from the *IRP_Client/Personal* folder, and the Identity will be removed from the Personal Identity ListView.

Personal Identities - Delete Identity – From IRP Server

If you right click anywhere on the Personal Identities ListView and select *Delete Identity – From IRP Server*, you will see the following dialog (similar to importing an identity from the IRP server):



UserID Filter:

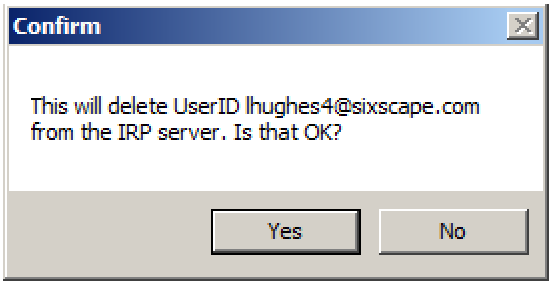
User Name filter: Hughes

User ID	User's Name
bhughes@sixscape.com	Bronwen Hughes
dhughes@sixscape.com	Dylan Hughes
lhughes@hughesnet.org	Lawrence E. Hughes
lhughes@sixscape.com	Lawrence Hughes
lhughes4@sixscape.com	Larry Hughes
mhughes@sixscape.com	Margaret Hughes
rhughes@sixscape.com	Remy Hughes

Select User and click OK

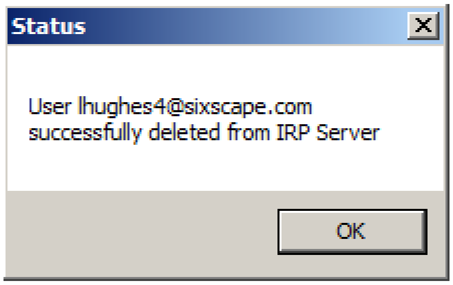
Enter a string in either or both filter boxes if you need to reduce the number of identities returned from the IRP server, and then select one identity by clicking on it, and click *OK*.

You will be asked to confirm this action:



This will delete UserID lhughes4@sixscape.com from the IRP server. Is that OK?

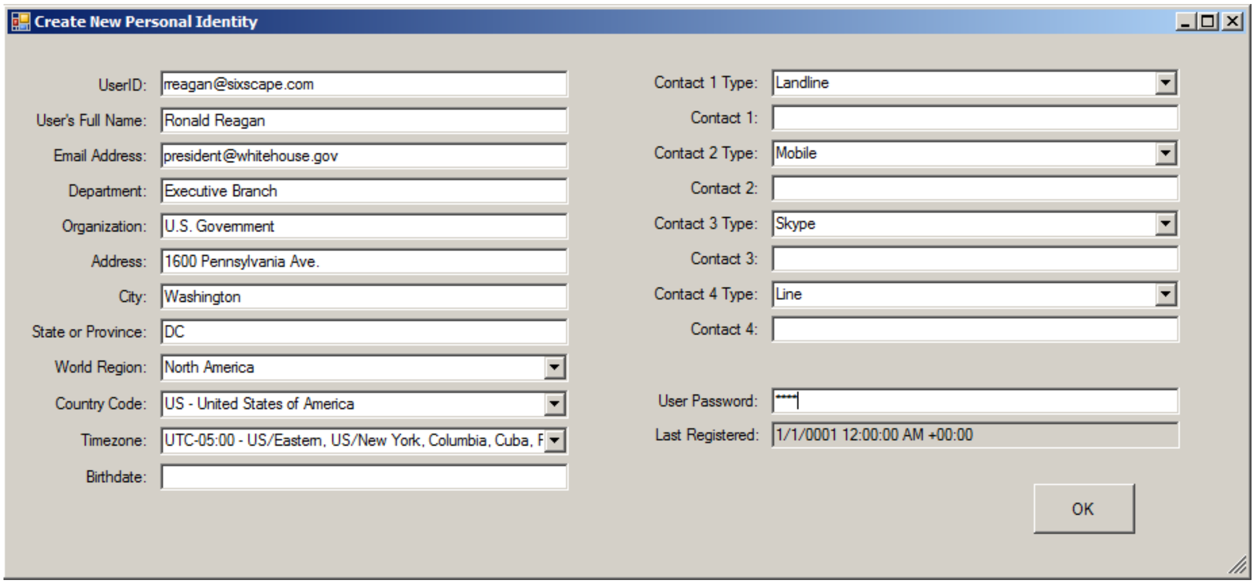
If you click *Yes*, it will send a *Delete_User* message to the IRP server. The result of this request will be displayed:



User lhughes4@sixscape.com successfully deleted from IRP Server

Personal Identities - Create New Identity

This allows you to create a new identity. Right click anywhere in the Personal Identity page and select *Create New Identity*. You will see the following dialog:



Create New Personal Identity

UserID:	rreagan@sixscape.com	Contact 1 Type:	Landline
User's Full Name:	Ronald Reagan	Contact 1:	
Email Address:	president@whitehouse.gov	Contact 2 Type:	Mobile
Department:	Executive Branch	Contact 2:	
Organization:	U.S. Government	Contact 3 Type:	Skype
Address:	1600 Pennsylvania Ave.	Contact 3:	
City:	Washington	Contact 4 Type:	Line
State or Province:	DC	Contact 4:	
World Region:	North America		
Country Code:	US - United States of America	User Password:	****
Timezone:	UTC-05:00 - US/Eastern, US/New York, Columbia, Cuba, F	Last Registered:	1/1/0001 12:00:00 AM +00:00
Birthdate:			

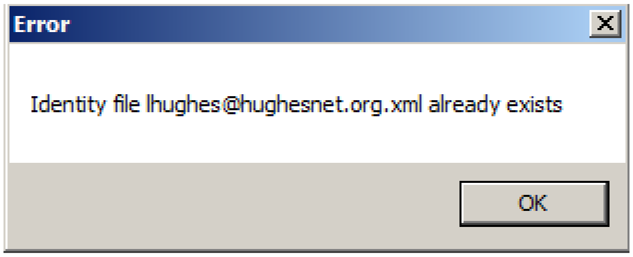
Enter details for the new Identity. You can always add more at another time by updating the identity. The only field that must be entered is the UserID.

The fields are the same as usual.

Normally a blank password field leaves the password unchanged. In this case, a blank password field will produce an empty password, and the user will not be able to login.

Click OK to create the identity.

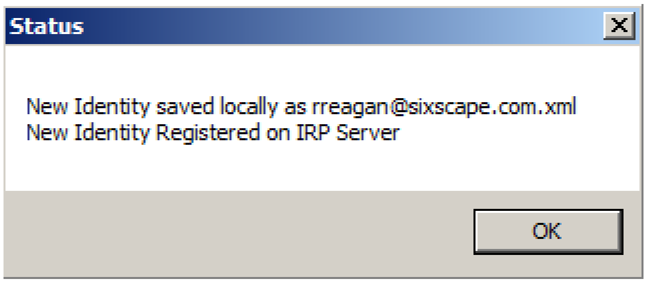
If the UserID is already in use locally, you will see the following:



Error

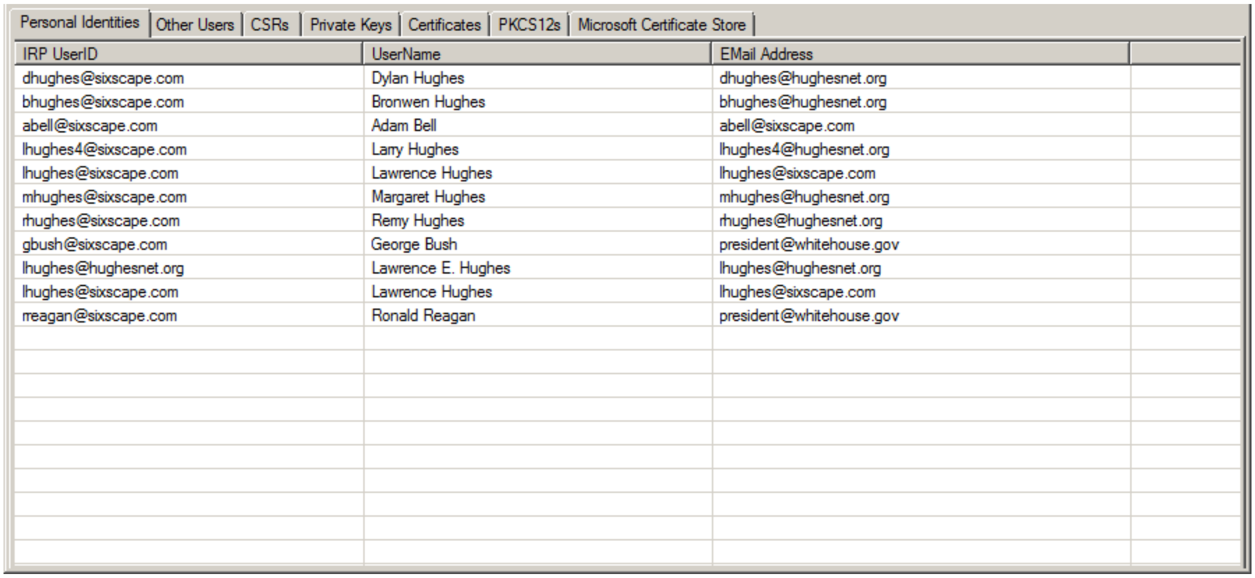
Identity file lhughes@hughesnet.org.xml already exists

If everything went OK, you will see the following:



New Identity saved locally as rreagan@sixscape.com.xml
New Identity Registered on IRP Server

And then the new identity will appear in the Personal Identities ListView:



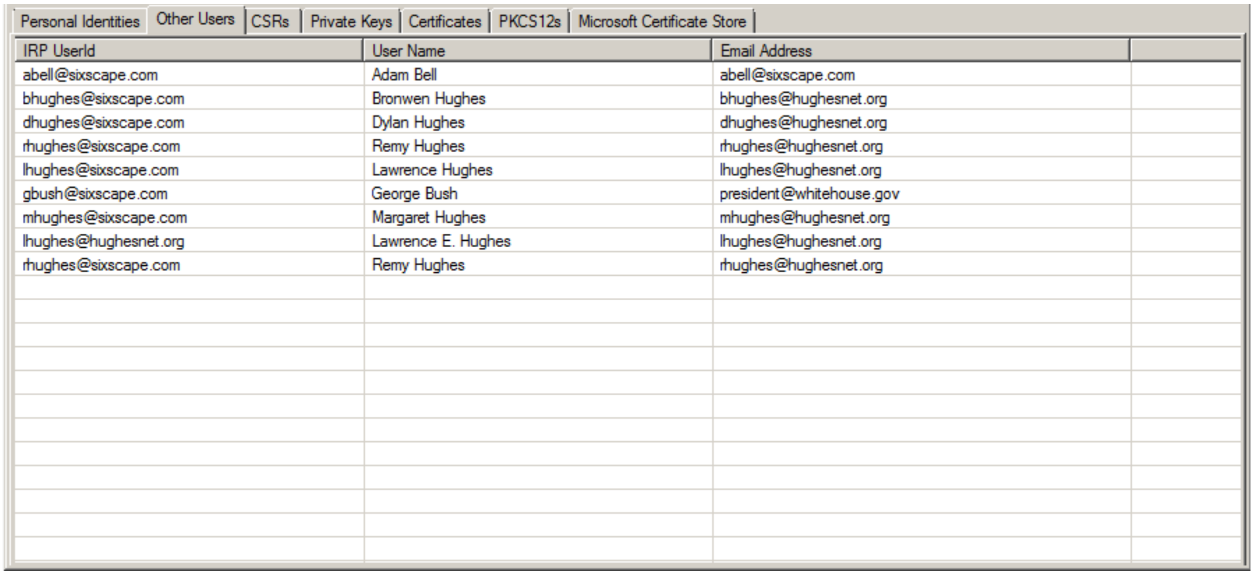
IRP UserID	UserName	EMail Address
dhughes@sixscape.com	Dylan Hughes	dhughes@hughesnet.org
bhughes@sixscape.com	Bronwen Hughes	bhughes@hughesnet.org
abell@sixscape.com	Adam Bell	abell@sixscape.com
lhughes4@sixscape.com	Larry Hughes	lhughes4@hughesnet.org
lhughes@sixscape.com	Lawrence Hughes	lhughes@sixscape.com
mhughes@sixscape.com	Margaret Hughes	mhughes@hughesnet.org
rhughes@sixscape.com	Remy Hughes	rhughes@hughesnet.org
gbush@sixscape.com	George Bush	president@whitehouse.gov
lhughes@hughesnet.org	Lawrence E. Hughes	lhughes@hughesnet.org
lhughes@sixscape.com	Lawrence Hughes	lhughes@sixscape.com
rreagan@sixscape.com	Ronald Reagan	president@whitehouse.gov

Tab Control - Other Users

The Others Users page is for downloading User Info and certificates for other people that you want to communicate with. You can search the IRP server for registered users and download their information. When you download their information, if they have uploaded a client cert that is downloaded as well.

[Coming soon: access to other users' information will be subject to fine grained access control lists.]

When you click on the *Other Users* tab, you see the Other Users' ListView:



IRP UserId	User Name	Email Address
abell@sixscape.com	Adam Bell	abell@sixscape.com
bhughes@sixscape.com	Bronwen Hughes	bhughes@hughesnet.org
dhughes@sixscape.com	Dylan Hughes	dhughes@hughesnet.org
rhughes@sixscape.com	Remy Hughes	rhughes@hughesnet.org
lhughes@sixscape.com	Lawrence Hughes	lhughes@hughesnet.org
gbush@sixscape.com	George Bush	president@whitehouse.gov
mhughes@sixscape.com	Margaret Hughes	mhughes@hughesnet.org
lhughes@hughesnet.org	Lawrence E. Hughes	lhughes@hughesnet.org
rhughes@sixscape.com	Remy Hughes	rhughes@hughesnet.org

This ListView looks and acts a lot like the Personal Identities ListView, with the following exceptions:

- You cannot create a new "Other" identity. You can only make changes to an Other User locally (these cannot be uploaded to the IRP server). You cannot delete an Other User on the IRP server
- Of the various PKI objects (CSR, private key, certificate, PKCS 12) you can only download, view, or use an Other User's certificate.
- You cannot use an Other User's identity for strong client authentication to a website.
- You can create a secure (signed and/or encrypted) to any of these users. To sign a message you must have obtained (and have present on your node) a private key and certificate for yourself. To send an encrypted message, you must have downloaded a certificate for the recipient.

If you right click, the context menu has the following items:

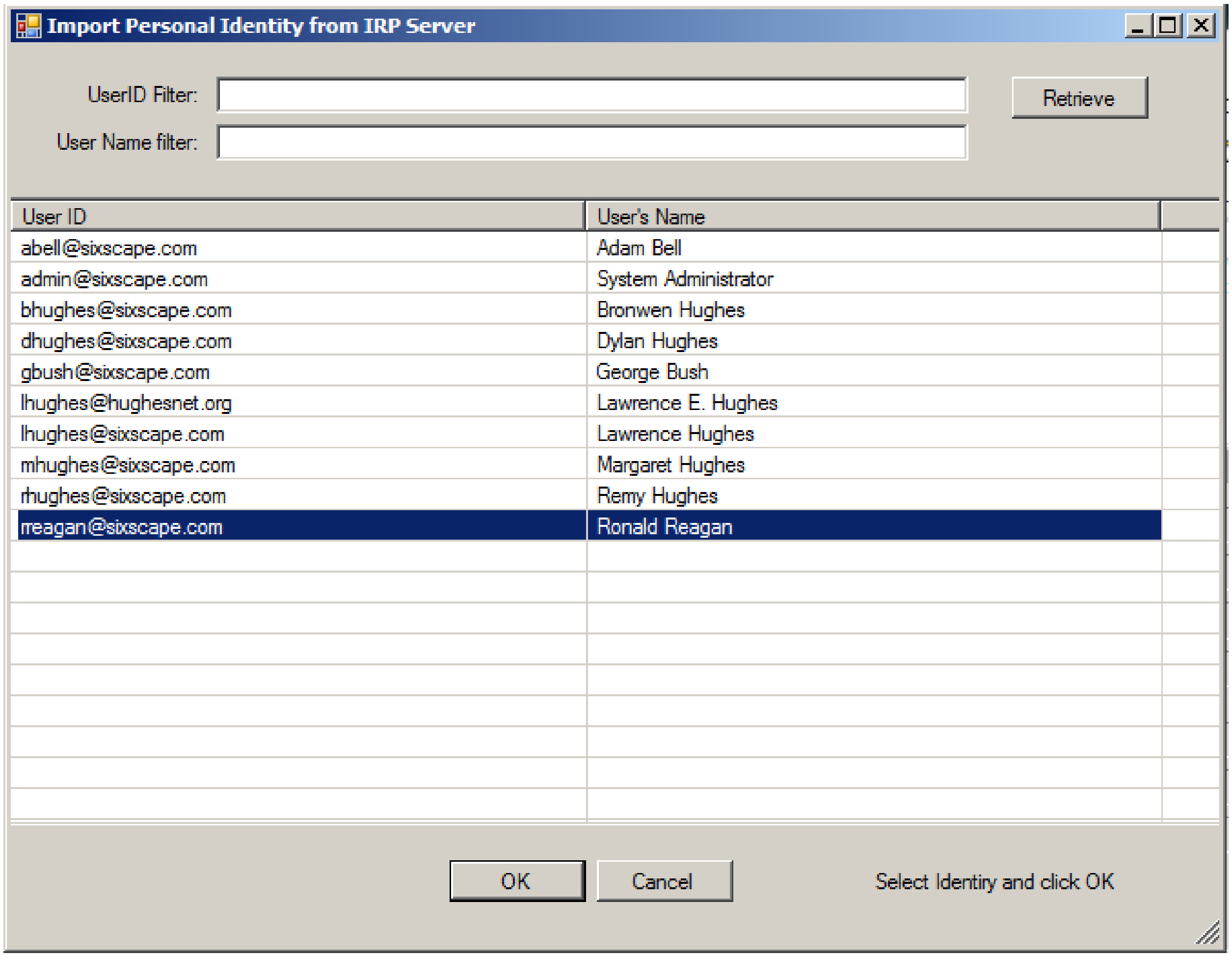
```
Import User from IRP

View User Info (same as double click)

Create Secure Message for User

Delete User Info
```

Other Users - Import User from IRP

This allows you to download any registered IRP user's *User Info* and client certificate (if they have uploaded one) from the IRP server. The downloaded information is kept in an .xml file (similar to the Personal Identity .xml file), in the *PKI_DB/Others* folder.

To import a user's info, right click anywhere in the Other Users ListView and select *Import User from IRP*.

You will see a list of users available to you. You can narrow the list by entering strings in the UserID and/or User Name filter GUI controls and clicking the *Retrieve* button. Only users whose User ID contains the specified UserID filter and User Name contains the specified User Name filter will be listed. A blank filter means "accept all".

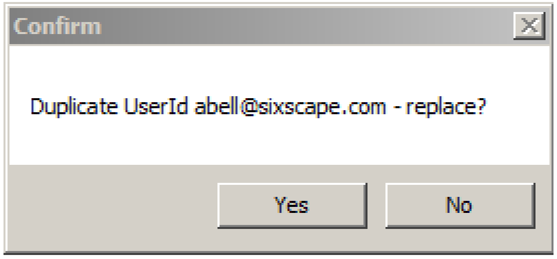


User ID	User's Name
abell@sixscape.com	Adam Bell
admin@sixscape.com	System Administrator
bhughes@sixscape.com	Bronwen Hughes
dhughes@sixscape.com	Dylan Hughes
gbush@sixscape.com	George Bush
lhughes@hughesnet.org	Lawrence E. Hughes
lhughes@sixscape.com	Lawrence Hughes
mhughes@sixscape.com	Margaret Hughes
rhughes@sixscape.com	Remy Hughes
rreagan@sixscape.com	Ronald Reagan

Select Identiry and click OK

Select an Identity to download by clicking on it, then click *OK*.

If the user already exists, you see the following message:

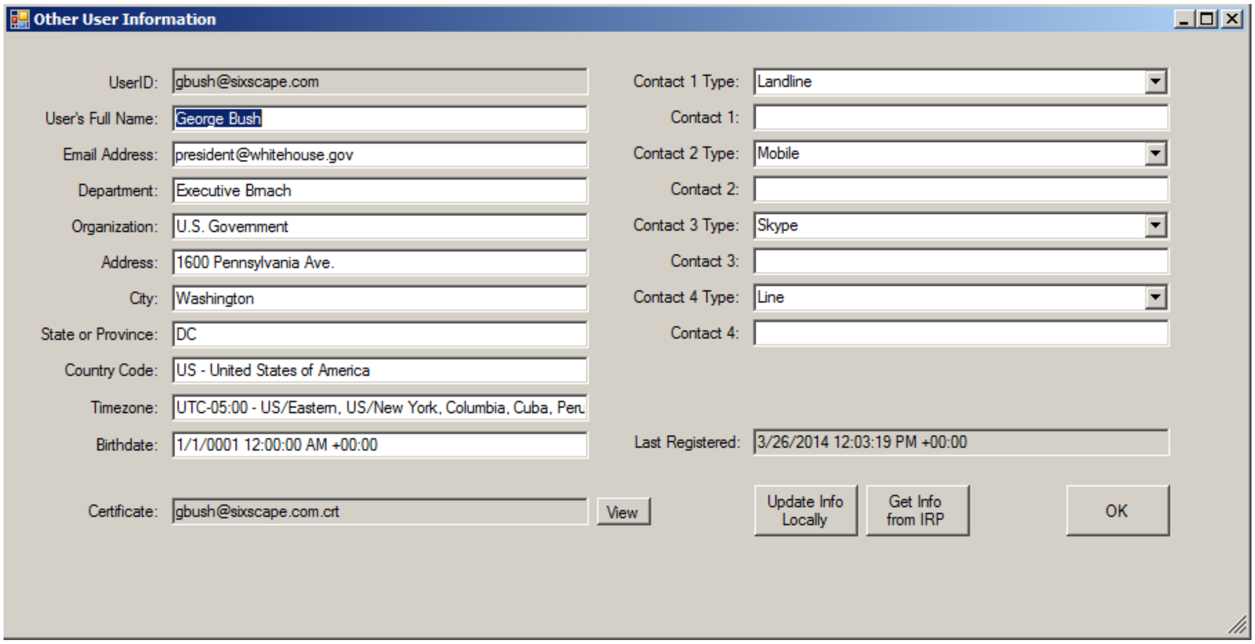


Duplicate UserId abell@sixscape.com - replace?

If you click *Yes*, then the user will be replaced with the new imported information. If you click *No*, the import will abort.

Other Users - View User Info

If you right click on a User and select *View User Info*, or if you double click on a User, you will see the following dialog box:



Other User Information

UserID: gbush@sixscape.com

User's Full Name: George Bush

Email Address: president@whitehouse.gov

Department: Executive Brnach

Organization: U.S. Government

Address: 1600 Pennsylvania Ave.

City: Washington

State or Province: DC

Country Code: US - United States of America

Timezone: UTC-05:00 - US/Eastern, US/New York, Columbia, Cuba, Peru

Birthdate: 1/1/0001 12:00:00 AM +00:00

Certificate: gbush@sixscape.com.crt

Contact 1 Type: Landline

Contact 1:

Contact 2 Type: Mobile

Contact 2:

Contact 3 Type: Skype

Contact 3:

Contact 4 Type: Line

Contact 4:

Last Registered: 3/26/2014 12:03:19 PM +00:00

This is similar to the Personal Identity dialog box, except that the information comes from an .xml file in the *IRP_Client/Others* folder, the only PKI object is the user's certificate, and fewer options are availble. Also, the Country Code and Timezone GUI controls are TextBoxes instead of ComboBoxes.

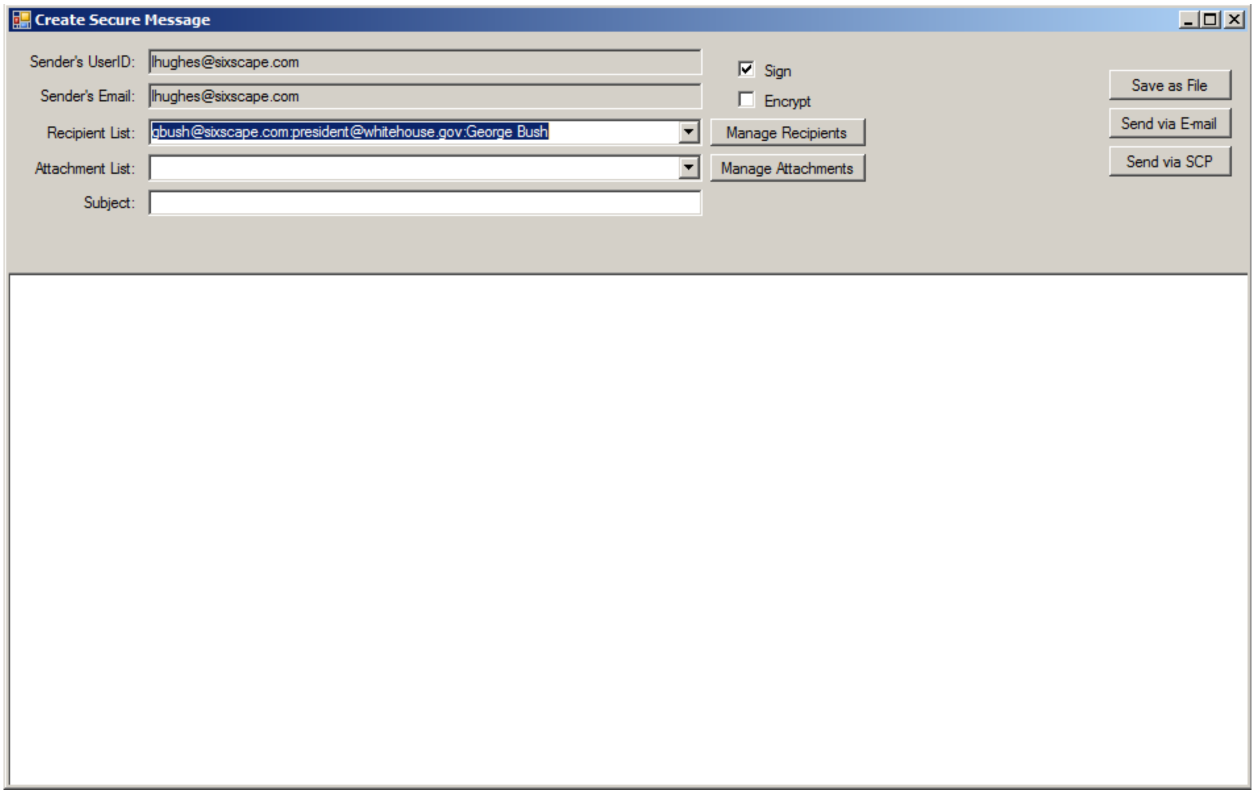
When you download user information, if that user's certificate exists on the server, it will also be downloaded to your IRP client. You can view this cert with the sole *View* button. Viewing a certificate works the same was as before.

The *Update Info Locally* button allows you to make changes to your local copy of the information (in the .xml file in *IRP_Client/Others* folder), such as a new phone number. However, there is no way to change the information on the IRP server. Also, if you click *Get Info from IRP*, it will overwrite any local updates with the most recently registered information from the IRP server for this user.

The *OK* button dismisses the dialog without doing anything else.

Other Users - Create Secure Message for User

If you right click on a user and select *Create Secure Message for User*, this will bring up an email compose window, with the currently logged-in user's name and email address as the sender, and the selected user as the first recipient:



Sender's UserID: lhughes@sixscape.com

Sender's Email: lhughes@sixscape.com

Recipient List: gbush@sixscape.com:president@whitehouse.gov:George Bush

Attachment List:

Subject:

☑ Sign

□ Encrypt

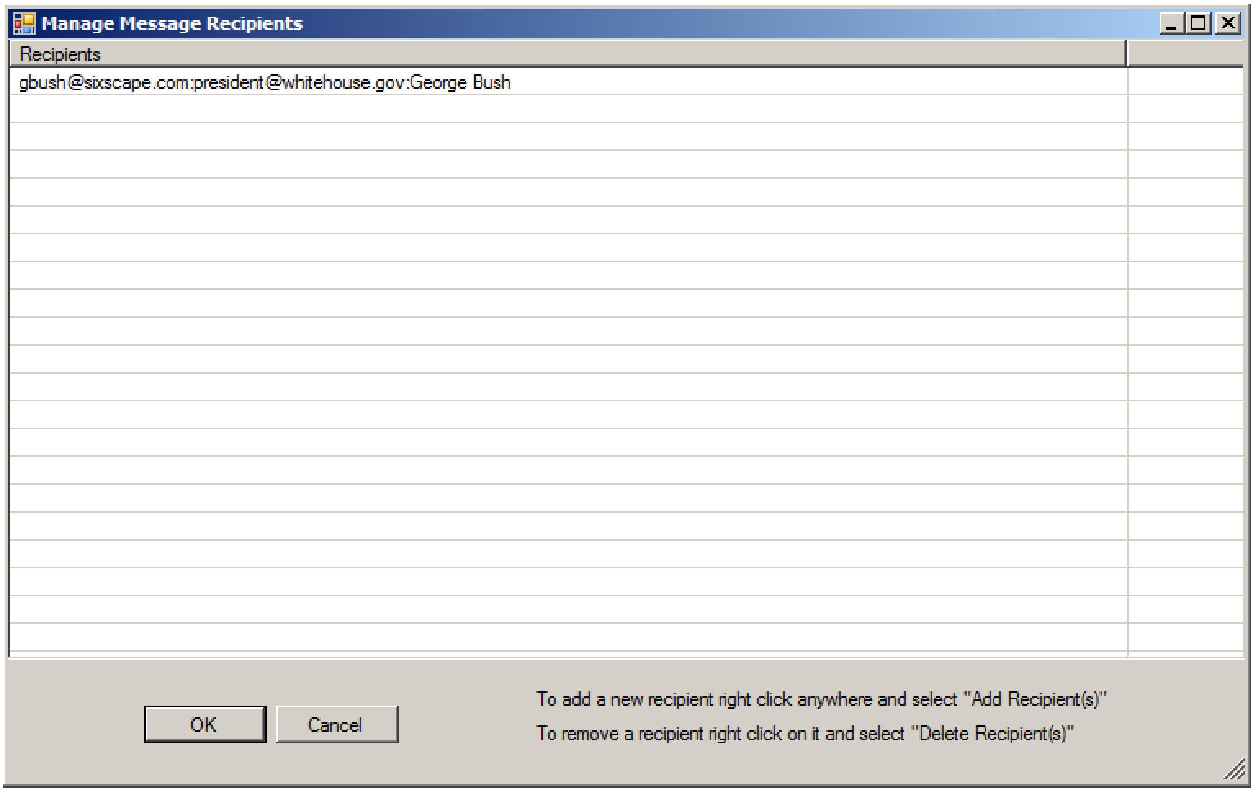
You can add additional recipients using the *Manage Recipients* button, and attachments using the *Manage Attachements* button. You can add a Subject and message body (the large window at bottom), and choose any combination of "Sign" and "Encrypt" (neither, either or both). Once done you can save the file in S/MIME format (with type ".eml") anywhere you like (for example in your DropBox or OneBox folders). Any S/MIME compliant email client (e.g. Windows Live Mail or Outlook) can load a secure .eml file created this way, and view or save attachments, etc.

The first time you try to create a secure message (after a new login), it will load your PKCS12 file so that you can sign the messages. Enter the passphrase for your PKCS12 file and confirm it. Once this is done, you can create any number of secure messages for any recipient(s) without reloading your PKCS12 file.

[Coming soon: you will be able to send the composed secure message via SMTP, or transfer it to an SSH compliant File Server like with scp]

Other Users - Create Secure Message - Manage Recipients

If you click on the *Manage Recipients* button, you will see the following dialog:



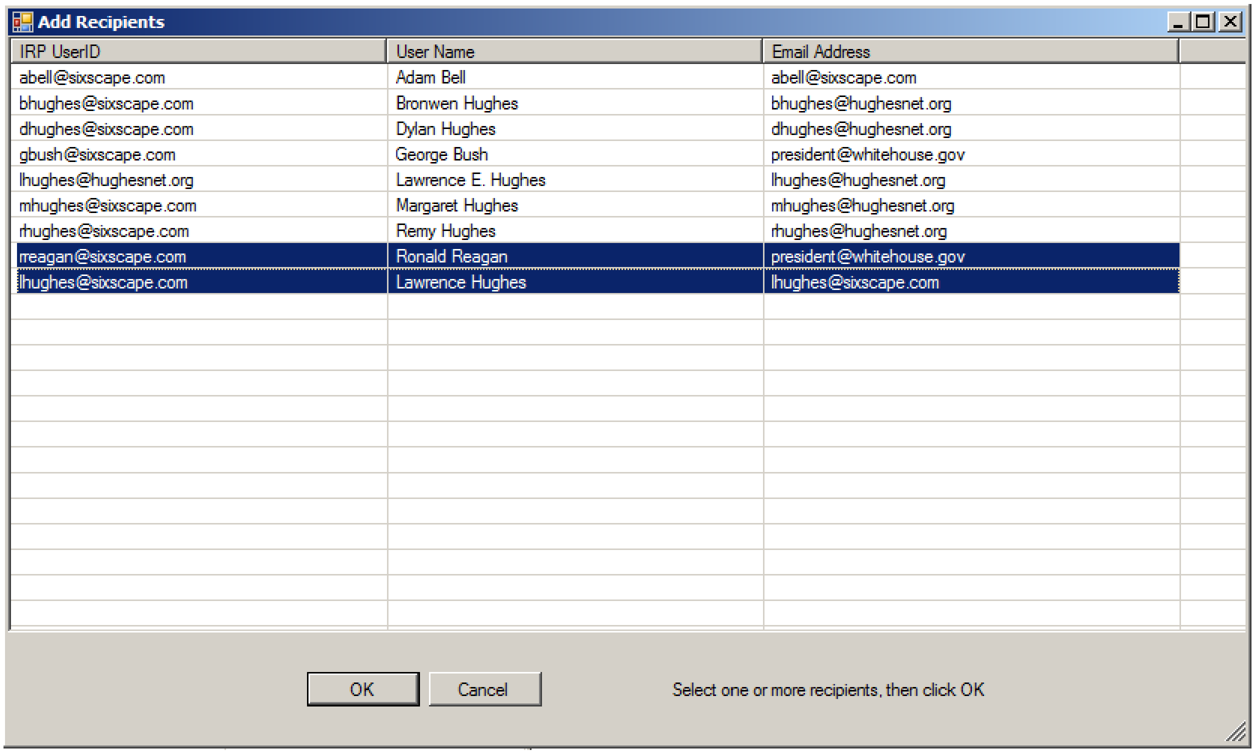
gbush@sixscape.com:president@whitehouse.gov:George Bush

To add a new recipient right click anywhere and select "Add Recipient(s)"

To remove a recipient right click on it and select "Delete Recipient(s)"

The current recipient(s) are displayed (UserID:email address:Name). To add a new recipient, right click anywhere in the ListView and select *Add Recipient(s)*. To delete one recipient, right click on it and select *Delete Recipient(s)*. To delete multiple recipients, hold down the ctrl key and click on any number of recipients and then right click on any of them and select *Delete Recipient(s)*.

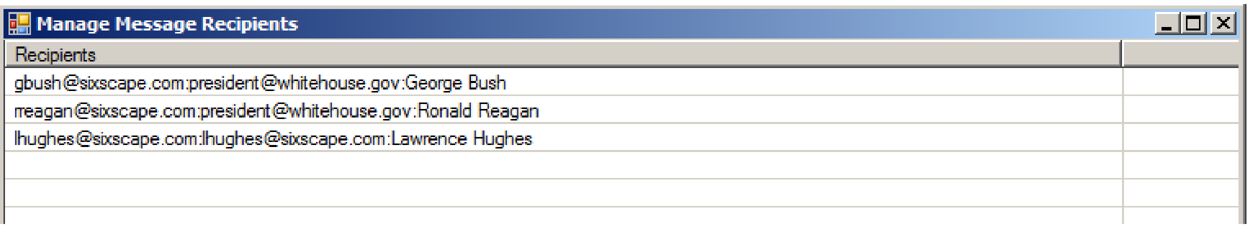
To add additional recipients, right click anywhere on the ListView control and select *Add Recipient(s)*. You will see the following dialog:



IRP UserID	User Name	Email Address
abell@sixscape.com	Adam Bell	abell@sixscape.com
bhughes@sixscape.com	Bronwen Hughes	bhughes@hughesnet.org
dhughes@sixscape.com	Dylan Hughes	dhughes@hughesnet.org
gbush@sixscape.com	George Bush	president@whitehouse.gov
lhughes@hughesnet.org	Lawrence E. Hughes	lhughes@hughesnet.org
mhughes@sixscape.com	Margaret Hughes	mhughes@hughesnet.org
rhughes@sixscape.com	Remy Hughes	rhughes@hughesnet.org
rreagan@sixscape.com	Ronald Reagan	president@whitehouse.gov
lhughes@sixscape.com	Lawrence Hughes	lhughes@sixscape.com

Select one or more recipients, then click OK

You can select one or more additional recipients by clicking on them. When done selecting recipients, click the *OK* button. The selected recipients will be added to the recipient list.

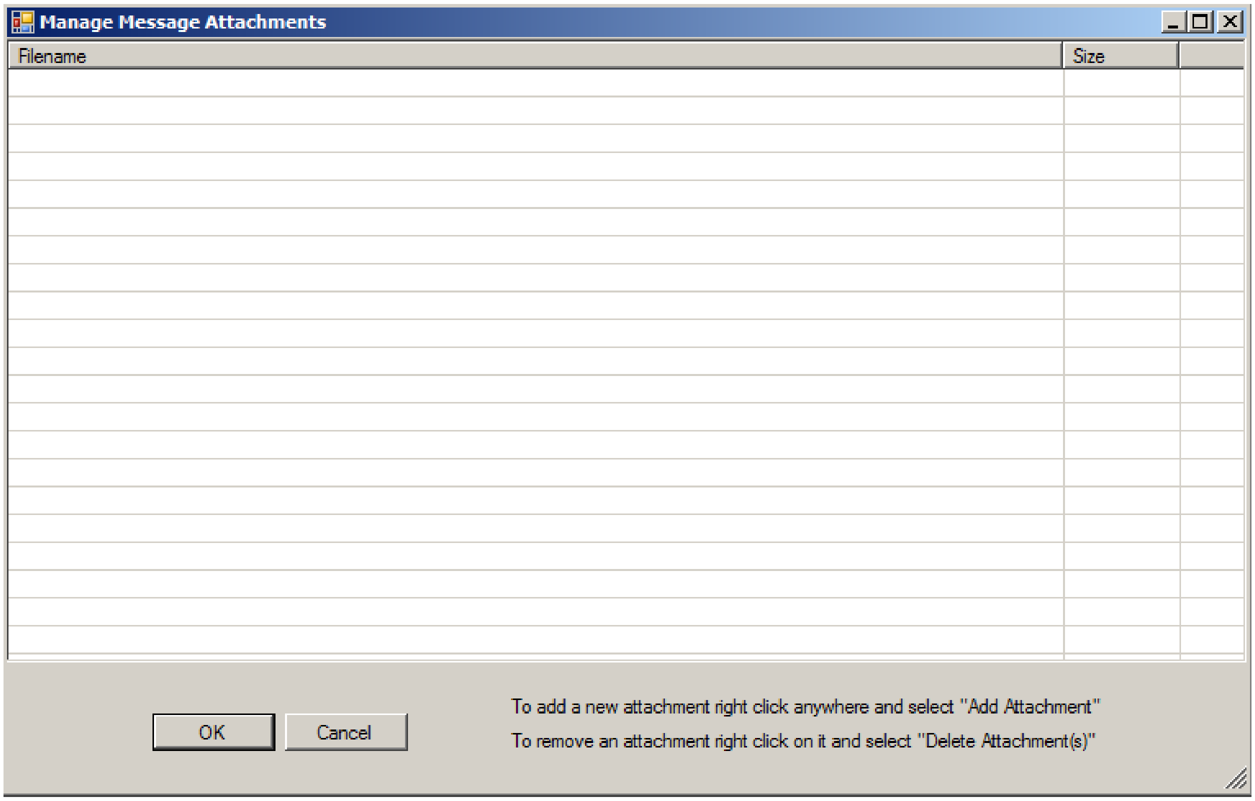


Recipients
gbush@sixscape.com:president@whitehouse.gov:George Bush
rreagan@sixscape.com:president@whitehouse.gov:Ronald Reagan
lhughes@sixscape.com:lhughes@sixscape.com:Lawrence Hughes

If you choose to encrypt the message, each recipient should have a client certificate in your IRP client. If you choose to sign the message, the current logged in user must have a client certificate and private key. If you want to be able to open an encrypted message, you should include your own account as one of the recipients (otherwise only the recipient will be able to open it). On return to the compose window, all recipients will appear in the pulldown list of the *Recipient List* combo box (although normally only the first one will display).

Other Users - Create Secure Message - Manage Attachments

If you click on the *Manage Attachments* button, you will see the following dialog:



To add a new attachment right click anywhere and select "Add Attachment"
To remove an attachment right click on it and select "Delete Attachment(s)"

To add a new attachment, right click anywhere in the ListView and select *Add Attachment*. You will see an “Open File” dialogbox that lets you select a file. When you click *Open*, the selected filename will be added to the list of attachments shown in the *Manage Message Attachments* ListView.

To delete an added attachment, right click on it and select *Delete Attachment*. You can also select several added attachments to delete and then right click on any of them.

Once you have the list of files to attach, click *OK*. As with recipients, the files to be attached will be shown in the *Attachment List* ComboBox (but only the first will be shown without expanding the pulldown list).

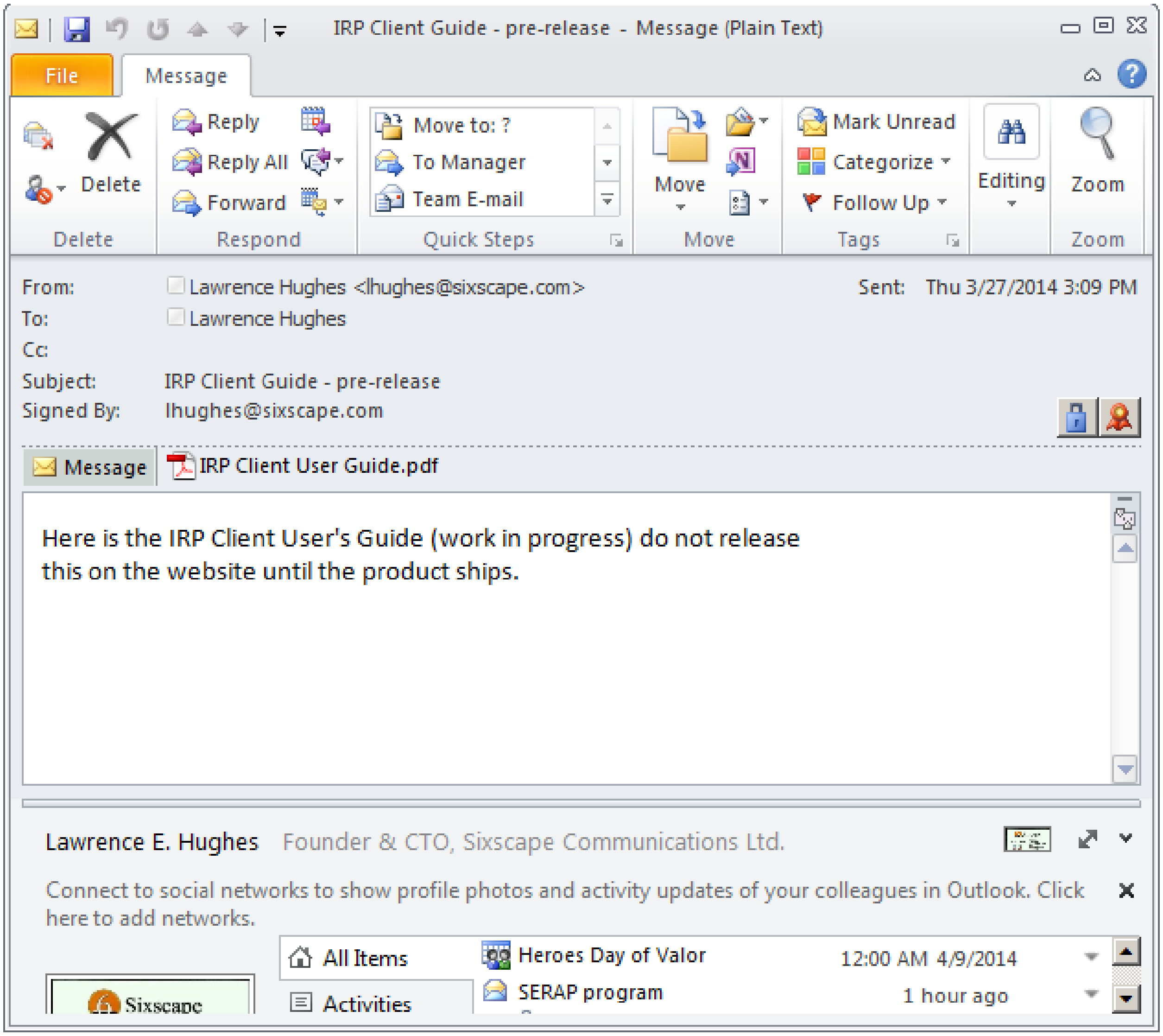
Other Users - Create Secure Message - Save as File

This option causes an S/MIME compliant message to be created from the sender, the recipient(s), the subject, the message body and any attachments, and written to a disk file. You will see a *Save File Dialog*. Select the directory and filename for the file to be written to. You can select DropBox or OneBox if you want to send secured information via those channels. The default filetype for an email message is ".eml".

You can open an email message by double clicking on this file (or right click and select "Open With..."). Most email clients (e.g. Outlook, Windows Live Mail, etc) are capable of opening email message files. The result is the same as if the file had arrived from an email server (it winds up in your message reading window). This allows you to view the message, save the attachments, etc. Your email client must have the recipient's IRP private key (and CA Certs) installed in its certificate store for this to work correctly. These are normally installed as needed in the normal operation of the IRP client (for Microsoft products). If you would like to use Netscape descended products (e.g. Thunderbird) you will currently need to export your key material in a PKCS12 format (using IRP client or your IE browser), then import that into the Mozilla certificate store. One way to do this is to make Firefox your default browser, then just double click on the PKCS 12 file. You will also need the CA certs for your IRP hierarchy installed. This is done in a similar manner, except that you export them in PEM format, not PFX.

[Coming soon: IRP Client will be able to import, export and use certificates in the Mozilla Certificate Store in the same manner it can do with the Microsoft Certificate Store currently.]

Typical result from opening a signed and encrypted message file (to myself) using Windows Live Mail:



From: Lawrence Hughes <lhughes@sixscape.com>
Sent: Thu 3/27/2014 3:09 PM
To: Lawrence Hughes
Cc:
Subject: IRP Client Guide - pre-release
Signed By: lhughes@sixscape.com

Here is the IRP Client User's Guide (work in progress) do not release this on the website until the product ships.

Lawrence E. Hughes Founder & CTO, Sixscape Communications Ltd.

Connect to social networks to show profile photos and activity updates of your colleagues in Outlook. Click here to add networks.

All Items
Activities

Heroes Day of Valor 12:00 AM 4/9/2014
SERAP program 1 hour ago

If you click on the Red Ribbon, it will show you information about the digital signature. If you click on the blue padlock, it will show you information about the encryption. If you double click on the .pdf file, you can read or save the document. The .eml file itself is secure and can be sent via insecure channels, saved for future reference, etc. You will need the private key of at least one of the recipients to open it.The public key of the signer is included, so the signature can always be checked.

The first few lines of the .eml file look like this:

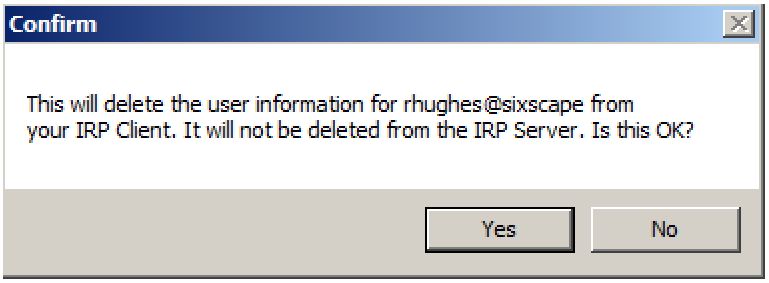
```
From: "Lawrence Hughes" lhughes@sixscape.com
To: "Lawrence Hughes" <lhughes@sixscape.com>
Content-Type: application/pkcs7-mime;
      name="smime.p7m";
      smime-type=enveloped-data
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
      filename="smime.p7m"
MIME-Version: 1.0
X-Mailer: EldoS MIMEBlackbox, version: 11.0.242.0
Date: Thu, 27 Mar 2014 15:08:39 +0800
Message-ID: <201403271508390886@17456350>
Subject: IRP Client Guide - pre-release
```

```
MIAGCSqGSIb3DQEHA6CAMIACAQAxggHOMIIBygIBADCBsTCBmzELMAkGA1UEBhMC
UEgxDTALBgNVBAgTBENlYnUxEjAQBgNVBAcTCUNlYnUgQ2l0eTEmMCQGA1UEChMd
U2l4c2NhcGUgQ29tbXVuaWNhdGlvbnMsIEx0ZC4xFzAVBgNVBAsTDlBLSSBPcGVy
YXRpb25zMSgwJgYDVQQDEx9TaXhzY2FwZSBDbGllbnQgQ0EgSW50ZXJtZWRpYXRl
AhEAgJBUIrlwf5Ow/T9eDeqJIzANBgkqhkiG9w0BAQEFAASCAQDXZOkf2O0G46Bq
EwJ49eo+WpnNPlSpbHVzER81JseGJcGVTA+BETxiyA1beoUbPbW4vqvwr1pD/EcC
lTif2yC5zVlfLkEthICJc79+gebl6TwMR8tZE0YMfPLndf4W+5JKZelJVOnJ3oCl
6QavK1MzX4EpfHcyRoEseqPsNE8dtRrIU2NdHmhsKUOKyYRVfICxraWMLhZikUqh
brp7XoB8PWa9JS9wA7yX9853XoeP/7RxjYbsxPN44+1VrljZzQgCypuSU7e9IZNU
u/DnPjmeV8dc3jZB3J5jeMvTSMDkp4Wy/k7JlL6tFsz96lvaespC15aIj+64bVnR
eIi5pSavMIAGCSqGSIb3DQEHATAUBggqhkiG9w0DBwQIZz3Keo7bnvaggASDISCQ
guIH2OSnHP1xP+NkcrDcw67vSEe0e4NoPXMcwt66nOHTUHJhjklp4KGAS6mUzyaD
0LNKrlcQACtl41L/rAJbUOXivjsqlArOaSE9eAxbN7UAj86LKESOZUArh2cWkpGJ
iAJXNemUR0M5foBQTBdHbQSy4QV1AtMCFWrdHQeMoA40v0J0Ikn+nkvA85zkZlvG
```

Other Users - Delete User Info

This option allows you to delete one of your “Other” users. It removes the local .xml file (from the *IRP_Client/Others* folder) and removes the name from the ListView. It does not affect the copy on the server in any way. You can always re-download the user from the server.

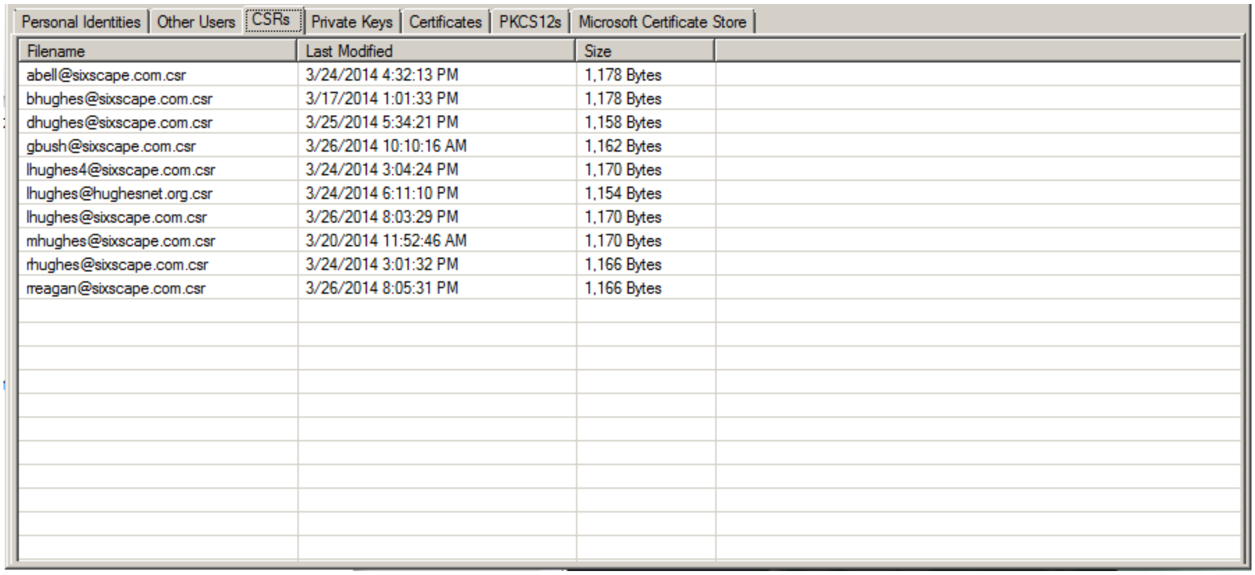
To delete a user, right click on a user in the ListView and select *Delete User Info*. You will see this dialog:



This will delete the user information for rhughes@sixscape from
your IRP Client. It will not be deleted from the IRP Server. Is this OK?

If you click *Yes*, the selected user will be removed from the ListView. If the user had a certificate it is not removed.

Tab Control – CSRs



Filename	Last Modified	Size
abell@sixscape.com.csr	3/24/2014 4:32:13 PM	1,178 Bytes
bhughes@sixscape.com.csr	3/17/2014 1:01:33 PM	1,178 Bytes
dhughes@sixscape.com.csr	3/25/2014 5:34:21 PM	1,158 Bytes
gbush@sixscape.com.csr	3/26/2014 10:10:16 AM	1,162 Bytes
lhughes4@sixscape.com.csr	3/24/2014 3:04:24 PM	1,170 Bytes
lhughes@hughesnet.org.csr	3/24/2014 6:11:10 PM	1,154 Bytes
lhughes@sixscape.com.csr	3/26/2014 8:03:29 PM	1,170 Bytes
mhughes@sixscape.com.csr	3/20/2014 11:52:46 AM	1,170 Bytes
rhughes@sixscape.com.csr	3/24/2014 3:01:32 PM	1,166 Bytes
rreagan@sixscape.com.csr	3/26/2014 8:05:31 PM	1,166 Bytes

This tab page allows you to view all Certificate Signing Requests (CSRs) currently on your system. This is basically just a directory of files in folder *IRP_Client\CSRs*. complete with filename, last modified date and file size.

When you request a certificate (Personal Identities – Request a Certificate), a CSR file is created and uploaded to the IRP server. The local CSR file is created in (and remains in) the *IRP_Client\CSRs* folder. You could submit the same CSR again, but for the most part you do not need a CSR once the corresponding certificate has been issued and downloaded. This interface is primarily for completeness. You could use it to check when you submitted a CSR if it hasn't been issued yet.

The contact menu in this ListView has the following options

```
Create CSR(and Key)

Import CSR (and Key)  – From File

                      - From IRP Server

Export CSR(and Key)   - To File

View CSR              (same as double click)

Delete CSR (and Key)  - From IRP Client

                      - From IRP Server
```

CSRs – Create CSR (and Key)

Normally you would request a certificate from the Personal Identities page. This option provides another way to generate a CSR.

[Coming soon: this option is not currently implemented – will allow requesting Server Certs and IPSec certs in addition to Client certs]

CSRs – Import CSR (and Key) – From File

This option allows you to import a CSR into the IRP_Client from anywhere. Right click in the CSRs ListView and select *Import CSR (and Key) – From File*. You will see an Open File Dialog set up to import files of type *.csr*. Navigate to the directory and file you want to import. If the file is not of type .csr, change the filter to *All Files (*.*)*. Once you click *Open*, the specified file will be copied into the *IRP_Client/CSRs* folder, and appear in the CSRs ListView. Only valid CSR files can be imported this way. The CSR can be in either PEM or DER format. It will be saved in PEM format (without password protection).

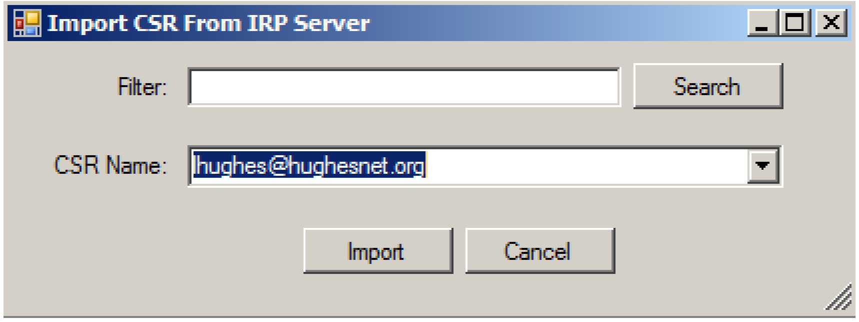
Assuming you successfully import a CSR, it will give you an option to also import a private key. This must be a valid private key file, in either PEM or DER format. It will be written to the *IRP_Client/PrivateKeys* folder, in PEM format. If the file is in PEM format, it probably is protected with a passphrase (which you will be prompted to enter). DER files are not protected with encryption. The IRP_Client will create a random password that protects the private key while in storage on the IRP_Client, and keeps it in the database. Anytime you retrieve the private key, the database passphrase is used to decrypt it.

Once a CSR has been imported, it will appear in the CSRs ListView, with last modification date and file size.

[Coming soon: the passphrase in the database will be encrypted – details to be decided]

CSRs – Import CSR (and Key) – From IRP Server

This option allows you to download a CSR (and optionally key) from the IRP server. You will see the following dialog:



Import CSR From IRP Server

Filter:

CSR Name: lhughes@hughesnet.org

You will only see CSRs for which the currently logged in identity is the "owner" (i.e. ones uploaded by that identity). Chose the one you want to download using the CSR Name ComboBox and click *Import*.

If there are too many CSRs listed, you can limit the number of them by entering a filter and clicking *Search*. Only CSRs that contain the entered string as a substring will be listed.

The CSR will be downloaded into the *IRP_Client/CSRs* folder, and immediately appear in the CSRs ListView.

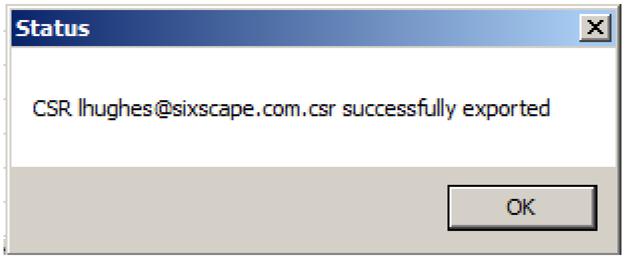
Despite the menu item name, unlike Import from File, it will not ask if you want to import a private key from the IRP Server. Usually private keys are not kept on the IRP server. If you really need to import a private key from the IRP server that can be done on the *Private Keys* tab.

CSRs – Export CSR (and Key) – To File

This option allows you to export a CSR (and optionally private key) from the IRP_Client to a file. The CSR will be saved in PEM format (without encryption). If you export the private key, you will need to specify the passphrase of the stored private key, then specify a new passphrase to protect the exported private key (which will be in PEM format).

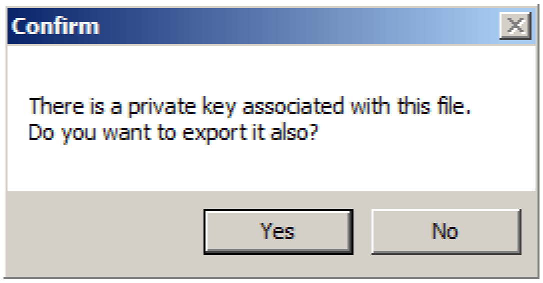
You will see a *Save File Dialog* set up to export a file with file type *.csr*. If you want to use a different file type, choose the “All Files (*.*)” filter in the Save File Dialog and enter your desired filetype.

If the CSR is successfully exported, you will see the following status box:



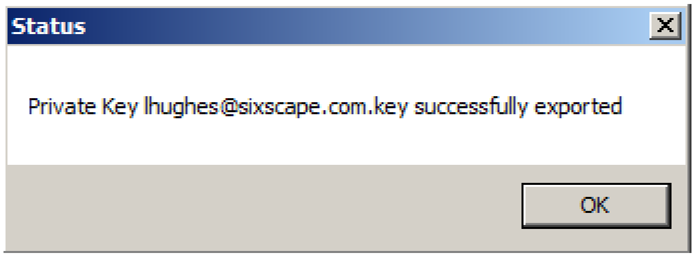
CSR lhughes@sixscape.com.csr successfully exported

Once the CSR is exported, if there is a private key associated with the CSR, it will ask you if you want to also export that as well:



There is a private key associated with this file.
Do you want to export it also?

If you click *Yes*, it will prompt you to enter the passphrase for the existing private key. Enter and confirm it, then click *OK*. You will then see another *Save File Dialog* (set up for file type *.key*). Finally you will be asked for another passphrase to protect the exported private key. Enter and confirm it then click *OK*. The private key will be written in PEM format encrypted with the second passphrase to the directory and file chosen in the *Save File Dialog* for the key. You will then see the following:



Private Key lhughes@sixscape.com.key successfully exported

[Coming soon: if you specify the export passphrase as an empty string, or click *Cancel*, the private key will not be encrypted. This can be useful for exporting a private key for a server cert, with Apache].

CSRs – View CSR

This option allows you to view the selected CSR (right click on one and select *View CSR*, or simply double click on any CSR). For details, see the section on viewing PKI objects.

Delete CSR (and Key) – From IRP Client