

Gedcom CGI Protocol and Web Services Family History and Technology Workshop 2005

John Finlay

PhpGedView Project Manager

Introduction

The Internet holds great potential for interconnecting different genealogy systems. Other industries have tapped into this potential by offering a variety of B2B and P2P services. For example, it is a simple matter to retrieve book information from Amazon or calculate shipping costs through UPS. However, genealogy systems remain isolated—neglecting the great advantages that could be obtained through interconnectivity over the Internet. Even though the GEDCOM specification provides a standard for the encoding of genealogy information, we still lack a common communication protocol. This presentation will discuss in detail a protocol that genealogy systems can use to communicate with each other both through HTTP/CGI calls and through Web Services. This presentation will also show how the PhpGedView and GDBI Open Source projects have implemented this protocol and the benefits both projects have received from interconnectivity.

HTTP CGI Implementation

This protocol follows a client-server system architecture. The client communicates with the server by sending name-value pairs through HTTP GET or POST requests to the server. The server will respond with data and/or a success code. All genealogy data exchanged between the systems is encoded in the GEDCOM 5.5 standard because it is a lightweight specification and because it is ubiquitous among genealogy programs.

Every client request must specify an “action” value. The “action” determines what other variables also need to be sent by the client. The client initiates a session with the server by sending a “connect” action. The server will respond with a name value pair containing this client’s unique session identifier. The client may initiate an anonymous read-only session or an authenticated session by providing a username and password. As determined by the server, authenticated clients may receive greater access to private data or be allowed editing privileges. After the initial connection is created, the client must include this session identifier with every future request.

Once the client is connected, it may then query the server for data through the “search” or “soundex” actions and request Gedcom record through the “get” action. Authenticated clients can send an “update” action to update a gedcom record.

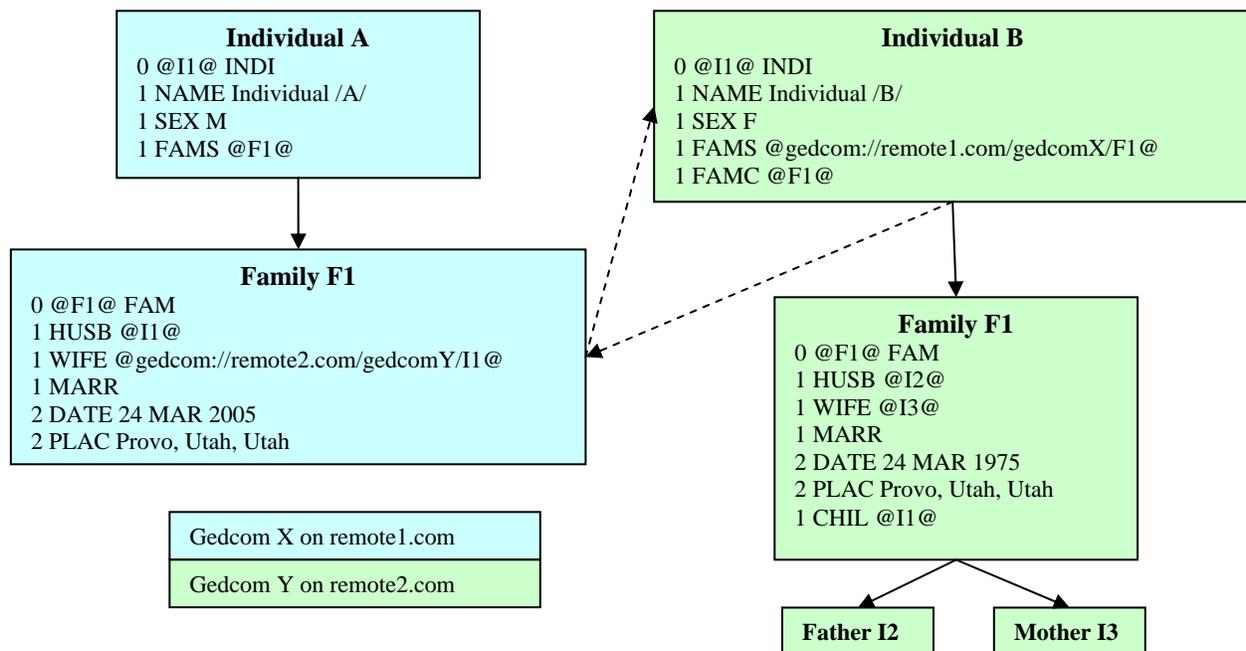
PhpGedView / GDBI Demonstration

PhpGedView is an Open Source project which provides an online environment for genealogy and implements the server side of this protocol. GDBI is an Open Source project which aims to provide an abstraction layer between a genealogy interface and a genealogy database. The GDBI project implements the client allowing remote users the ability to connect to a PhpGedView database and view and/or manipulate the database through a rich user interface. This interconnectivity benefits both projects as they work together to improve the technology systems for genealogists.

Using the protocol to Link Gedcoms

Something that is often requested is the ability to link people in different Gedcoms such that individual A in Gedcom X links to individual B in Gedcom Y. This would be very useful, for example, when trees are joined through marriages. We could even expand this across the internet and have Gedcom X and Gedcom Y could reside on different computers. The GEDCOM specification could easily be adapted to do this.

The following example illustrates how we might link two people together in this way:



In the example, you can see the Individual A is the husband of Family F1 and that the wife of Family F1 references Individual B through a gedcom:// URL. Just like http, jdbc, or other communications protocols, the gedcom:// URL would tell the system holding Gedcom X, that the wife in Family F1 resides on server “remote2.com” in the Gedcom “Gedcom Y” and has the Gedcom XREF ID of I1 in that Gedcom. It also tells the system that it should use the Gedcom protocol to retrieve the information the individual gedcom record.

Having retrieved the record for Individual B, the remote1.com server could then build a pedigree tree for Individual B by looking up the family record F1 on the remote2.com server by replacing 1 FAMC @F1@ with 1 FAMC @gedcom://server2.com/gedcomY/F1. After obtaining the family record, server1.com could then obtain Father I2 and Mother I3’s records in a similar fashion.

Anyone implementing these types of links would probably want to cache the records for performance and regularly check the remote server for updates. Some sort of trust would also need to be established between the two systems to ensure that privacy rules are maintained.

With this protocol and the simple extension of the GEDCOM specification, you can see how it would be possible to truly create that elusive world tree. It allows people to maintain control and ownership over their own data, but still allow it to be shared and linked into other family trees.

References

CGI GEDCOM Access Protocol <http://www.phpgedview.net/racp.php>

PhpGedView Project <http://www.phpgedview.net/about.php>

GDBI Project <http://gdbi.sourceforge.net>