FileMaker Info For Here or To Go!

FileMaker products help people connect to the world of data, and sync capabilities help people stay connected to their FileMaker solutions, wherever they are.

There are three main reasons why mobility is important to FileMaker developers: mobile users, multiple locations, and solutions to "other" technical needs. The third category includes some fun, complex motives for synchronisation, however most organisations initially use data synchronisation for the first two reasons.

FileMaker Info To Go: Mobile Users

Particularly true for road warriors, but any mobile user needs a copy of the database they use right on their laptops. They need to be able to work with the database at all times, even while offline. They need access to the same interface (i.e. layouts, scripts and reports) as users on the LAN at the "home" office. And they need it "right now," without waiting for the database to load and move data across the Internet.

FileMaker Info For Here and There: Multiple Locations

Whether the other locations include remote offices, or simply extend to a home office, the concept of one single physical data location is outdated. Still, in an Internet-accessible world, a Web application isn't always the answer either.

Synchronisation can provide the best of both worlds. Each end-user can have their own copy of the solution, utilising the Internet to keep their data in sync with other copies of the solution.

Want Fries with that?: Other reasons to sync

The same technology that can power the road warrior's data needs, and deliver data to and from multiple locations, can also solve other problems.

Hopefully your server will never fry. To minimise the impact, just in case, a back-up and recovery system, potentially to a warm-standby, lets companies protect mission critical data systems with continual, differential database backup so that nothing is lost or corrupted by a crash; or so a company can instantly switch to a standby server if one goes down. Synchronising data regularly ensures that users have the most up-to-date information.

Another reason to synchronise is data integration, which allows a company to share data between different systems and applications using intelligent, one-way and two-way, near-real-time replication services.

Yet another problem synchronisation can solve is web server replication to connect your hosted web site to your main systems and allow intelligent flow of data between each.

Also, synchronisation allows a company to collaborate with partners. Any partner collaboration requires a balance between data sharing and access control. The scenario requires a flexible approach to data ownership and distribution where control of data flow must also be distributed.

Taking It All Home: How to Sync FileMaker Solutions

As with any FileMaker solution, one can custom script a whole lot of functionality. Scripting is even more possible with an unlimited budget. While tempting, the complexities of data replication and synchronisation could best be achieved by those whose products contend with the issues on a daily basis.

Known for its technical depth and strength, one such product that provides FileMaker-ready sync capabilities is SyncDeK.

SyncDeK is a companion product to FileMaker database software that provides bi-directional, asynchronous data replication for data in FileMaker databases, which are either persistently connected to the Internet or used offline.

SyncDeK's technology lets developers extend FileMaker solutions to include mobile workers, remote offices, a constant backup, integration with SQL data sources, etc. SyncDeK-enabled solutions can synchronise database changes automatically or on demand. Your mobile users can have the most up-to-date information with them, plus they can update everyone else with their changes, even if they only have Internet access occasionally.

The specific technology used by SyncDeK was sufficiently unique that the U.S. Patent Office recently granted the company a patent for it. The patent covers SyncDeK's bi-directional asynchronous data replication technique for database management and synchronisation implemented across a peer-to-peer computer network.

Within a sharing community, database record changes are synchronised by storing change information in a synchronisation object, which is transmitted by a local-user computer at any time to a routing agent, which in turn transmits the synchronisation object at any time to a remote-user computer within the sharing community, and the synchronisation object is then used by that remote-user computer to update database records. Each synchronisation object is encrypted prior to transmission.

How Does SyncDeK Work?

At its core, SyncDeK is a background service that runs next to each copy of a FileMaker database that you distribute. Each of these SyncDeK Engines executes four key steps each time it performs a sync cycle. The descriptions below provide a very brief overview of each of these steps:

First, the SyncDeK Engine uses JDBC to query the local database to collect changes that have occurred since the last cycle. These collected records are processed into replication instructions and packaged into encrypted and compressed locally stored files.

Second, the SyncDeK Engine connects to a SyncDeK Server to upload the replication instructions for further processing and eventual distribution to other computers that share the database. SyncDeK Server is a background service installed on one computer within the network, or on an Internet Service Provider, such as dbDom.com.

Third, the SyncDeK Engine again connects to a SyncDeK Server, but this time to download replication instruction files that have been sent by other computers in the sharing community, and processed by SyncDeK Server for conflict resolution.

Finally, the SyncDeK Engine unpacks the downloaded replication instruction files and again uses JDBC to push the contents into corresponding local databases.

Each of these steps operates essentially independently of the others, and thereby has its own conditions of successful completion. In other words, there is a great deal of fault tolerance built into SyncDeK to cope with unexpected failures such as dropped Internet connections, database crashes or simply if a user chooses to close their computer and leave the coffee shop before a sync cycle completes. Additionally, a detail log of process activity is maintained at each node. This log is stored in XML format and is uploaded to SyncDeK Server at the end of each sync cycle. This provides a means for centralised monitoring of synchronization activity.

In addition to the SyncDeK Engine at each node, and to the SyncDeK Server used for distribution, there are a some other important elements which provide the means to configure and distribute a database solution to multiple computers. These are:

* Five fields pasted into each table to be synchronised;
* An Account added to each database that SyncDeK will use to make JDBC connections
* A plug-in for FileMaker that enables deletion notices to be sent to the local SyncDeK Engine as users delete records;
* A Web Viewer object pasted into one layout within your solution that provides a means to interact with the local SyncDeK Engine, Including providing setup instructions during initial configuration;
* and finally a folder called SyncDeK_Prefs, which gets built by the SyncDeK Engine based on configuration instructions provided through the Web Viewer object, and which contains a collection of XML preference files that the SyncDeK Engine uses to know which databases, tables and fields to track and how to connect to a designated SyncDeK Server.

For further information about how SyncDeK works and how to implement it within your solutions, see: http://worldsync.com/devguide.html.

####