FREEDOM DATABASE

Thursday, November 14, 2002

Introduction

> *Imagine all your staff, suppliers and customers – using different applications and databases –sharing data from a single managed source. With Freedom taken to it's logical extents – you can.*
>
> *Industrial designers and engineering can work in their preferred environment, accounts and logistics in theirs – management can oversee all these areas in virtually any style of view – spreadsheet, text report, graphical… anything.*
>
> *The raw elements used in your product can be sourced directly from the manufacturer's dataset… your corporate website and catalogues can be generated automatically from the common information base. The bottom line: you need never worry about duplicated or old information in your data again. We provide the information, you provide the ways to use it.*

Freedom is an innovative method utilising existing database back-ends to create an extremely powerful application oriented data infrastructure. Freedom may, in turn be extended to interface with existing desktop and enterprise software applications – to realise an information storage model that is unmatched in flexibility and application.

BENEFITS

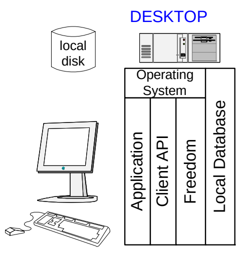
- Flexibility
- Retrofit to existing customers, applications and databases
- Ability to connect large, disparate data sets seamlessly
- Flexibility
- Opens extraordinary new opportunities for Business-to-Business data sharing
- Flexibility

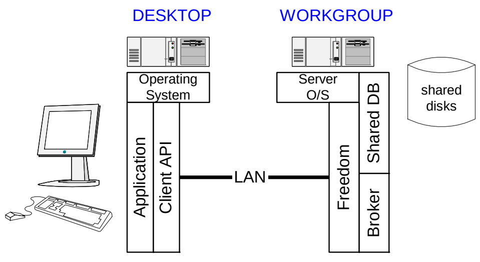
Overview

Freedom is not a database engine as such, but rather - a database schema that allows extraordinary flexibility while using generic products like MS-Access, SQL Server, Oracle etc. Freedom is not bound to any single database vendor, and can as it stands, co-operate simultaneously with multiple databases running on different platforms.

A long-term objective of the project is to develop a native Freedom 'database engine' which would exhibit significant performance gains, along with increased functional flexibility – while eliminating those resources consumed by the generic back-end products.

With this in mind, it is easiest to consider Freedom as an API or 'engine' resource that can exist at several levels - either as a local interface to a local database, a local interface to a remote database, or a remote interface to remote databases - or indeed - any combination of those topologies simultaneously.



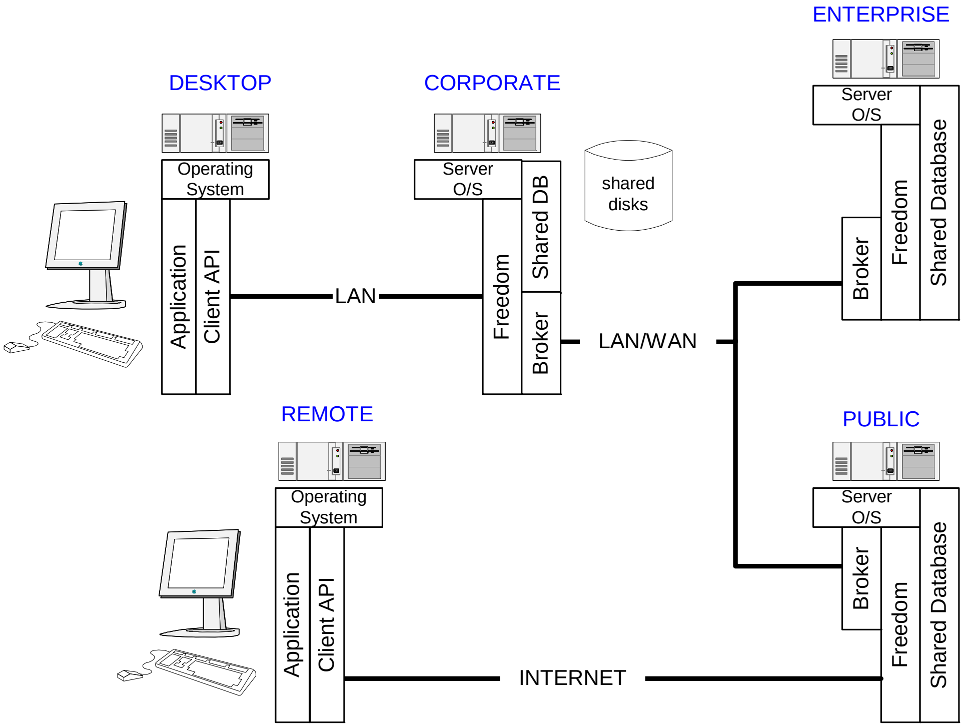


Freedom as a desktop/standalone solution, or in a workgroup environment.

The existing Freedom 'proof-of-concept' application encapsulates the first two modes of operation (local or remote data), and is written with the user interface tightly bound to the Freedom engine (simply a legacy of the way the platform evolved). The databases may be located anywhere there is sufficient bandwidth to sustain a reasonable flow of information between the sites.

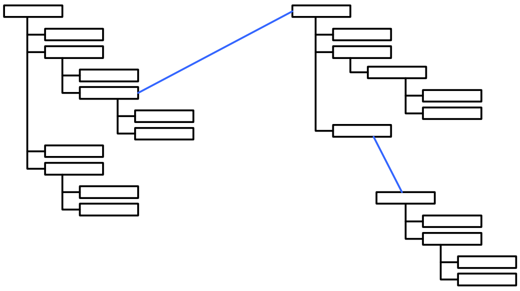
In the intended deployment model, Freedom would run as a system service on your computer, and be effectively invisible to the user. Macros, applets and complete applications would use the resources and data interface provided by Freedom - rather than having a direct database component within the application package.

This describes Freedom at the client level, however, the real power exists in using Freedom in a client-server (or more specifically... a distributed-server) model. Using a client-server architecture allows multiple users to interact with the same data at the same time, while viewing it in completely different paradigms... e.g. a designer collects various components to create a product in his preferred design tool, while the accountant sees real-time budgeting forecasts based on 'what-if' rule sets in Excel. In the warehouse, the logistics team can see dynamic stock requirements based on the latest approved design under their SAP CAM applications



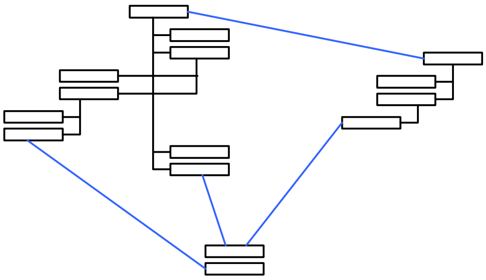
On the larger scale, Freedom data sets may be shared, mirrored,
or distributed in any preferred topology.

While much of this is already available in multi-level object-oriented and relational databases, Freedom extends it to the farthest logical limits. One-to-many, many-to-one, any-to-any… ! To traverse a path like that to be described in a moment in a conventional RDB or ORDB would require a substantial programming effort – possibly man months. In Freedom, it's almost effortless. An extremely simple programmer's interface - to create, manage and extract physical or inferred information from any data set.



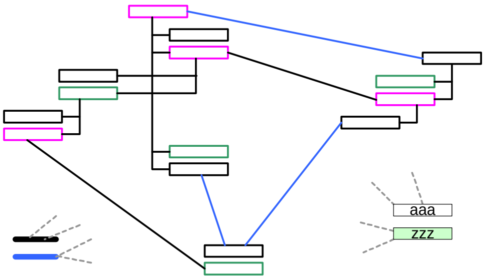
Traditional hierarchical databases
A lot of information, but very rigid paths

In a Freedom data collection, there is no rigid structure, you may organise, duplicate, inherit and cross-reference objects absolutely freely. The power of the database is not only in the data storage mechanism which is very unique to the Freedom concept, but is augmented by the use of 'attributes' which may be attached to any level of object, or object family.



Freedom allows data be organised by any
logical relationship. Paths between data and
datasets are defined as you need them.

This understates the power of the system - however suffice to say that it is quite practical to have your children's homework, your recipes, family tree, dog's vet history and corporate personnel and technical design documentation in the same Freedom data structure - completely isolated in relevance, but completely integrated where references are appropriate.



Freedom also allows any object to be independently classified as a member of a generic type - even while they are within a functioning data set.... these types may then have attributes and relationships of their own

With this super-database in place – exactly as it is today in the proof-of-concept demonstration - it is perfectly reasonable to maneuver the following information path...

The Universe
The Milky Way
The Earth
Australia
Victoria (where I live!)
A power station
A trunk power line
Trace to a substation

A local low-voltage circuit
Trace to My Street
My House
My Circuit Breaker
Trace to My AC outlet
Trace to My personal computer
Trace the license of the operating system :
me! < reseller < distributor < developer
Up to my son's personal information
Move up to my son’s instance
Trace to me (as his father, via a paternal birth relationship)

Jump to my ‘proxy’ at my place of employment

Jump to a design project I am involved with
Generate a complete documentation package for the project
Drill-down into a specific component
Review the supplier’s website or online database
Back to the component
Select an input connector
Trace back to the ultimate source of the input data to that pin
Evaluate the source object path
Select an intermediate element in the source path
Review an Acrobat product document for that device
Review / Set product specific attributes for that specific item
Review / Set product specific attributes for all items of that generic type

Trace back to the manufacturer's sales node using the purchase contract
Drill up to the 'About Us' node of that manufacturer
View images or other documents relevant to that organisation
:
:
And so-on - literally infinitely, across any connected datasets

And remember - every node we have passed may have attributes - as simple as 'date-of-birth', or coffee preferences, to automatically maintained values… calculated net-worth, local time, video clips, other databases... or literally any other data that can be stored or accessed with your computer.

If the data is accessible - and with permissions - you could find the name of the artisan that crafted the left propeller fitted to Bill Gate's favourite speedboat. Infinite breadth of relational information is possible - and the privacy of that information is in the hands of the person that puts the information into the database.

While Freedom on a single computer can provide enormous flexibility in sharing data among different paradigms – the real power comes with many people sharing knowledge in the same Freedom environment, hence a client-server model as discussed earlier is fundamental. Taking that concept a step further, allows ad-hoc connections between diverse servers simultaneously - using a 'broker' gateway to negotiate sharing of data across servers – irrespective of their physical location. Of course security and 'permissions are applied across the whole Freedom world - putting the security of every data item in the hands of it's creator/owner.

Taking the above dramatic example of traversing the Milky Way, through my son's PC, and across to a component in an apparently unrelated world - highlights the complexity encompassed by Freedom, yet exposed to you in a clear, organised, relevant manner - while not bounding you in any dimension. Combining that with the idea of 'brokers' - allows you to maintain 'live' online data systems of global information. Every element - of every item - within your Freedom data horizon - is available at the discretion of the data owner. Perhaps *you* can see the 'sell' prices, and whether or not they have enough stock for your enquiry, but 'cost' prices and actual stock quantities are not divulged at the manufacturer's option.

Using an API/engine metaphor - developers may create 'agent applications' to explore and create or update whole object trees, or simple attributes of selected records - based on specific criteria... (e.g. updating world time or 'age at next birthday') - completely transparently to every user - except that they see the right data in real-time, every-time. Data mining takes on a whole new meaning.

The above example highlights one of the reasons we developed Freedom - to eliminate multiple copies of the same data. An excerpt of a spreadsheet with change auditing, a typed phone list versus the copy on the intranet, remote managers changing site specific information while not advising head-office. All these can be minimised or completely eliminated with Freedom.

> *Even the idea of taking an Excel sheet home on a floppy disk (you can if you want to - but why not access Freedom over the internet?) Ok, you edit the floppy and take it back to work... ask your secretary to copy the changes back into the 'master sheet' - unaware that your office colleague is doing exactly the same thing !!!. Yes there is conflict resolution, but someone has to determine the precedence of changes.*

Because Freedom is a database (but can be hidden behind Excel or …), and every item of information has a known place in the data hierarchy... data replication and merge/conflict resolution is largely automatic - requiring user intervention only when the source and target data will fail specific user-defined validation rules. Duplicated out-of-sync information is a thing of the past.

Ok - that's enough for now - but suffice to say - with Freedom in place - virtually ANY user interface can be developed easily - that can your data on your computer in any form you desire.

Dynamic, self-modifying, global, intelligent applications are really possible.

To repeat the opening points – it really is as good as it sounds…

- Absolute flexibility
- Retrofittable to existing customers, applications and databases
- Ability to connect large, disparate data sets seamlessly - anywhere
- Real opportunities for B-to-B data sharing

M.D.Coop

freedom@SL4P.net