

## Servoy Comparative Functionality Guide for FileMaker® Pro

---

### About This Whitepaper

The purpose of this whitepaper is to help IT managers and FileMaker developers understand the fundamental differences of how Servoy works in terms of architecture, development methodologies, and integration techniques by comparison to FileMaker.

Although they share a few similarities in terms of development workflow, Servoy and FileMaker are completely different products – each with its own set of strengths and limitations. The key is to choose the right tool for the right job.

### Introduction

In comparing and contrasting the benefits and limitations of Servoy and FileMaker, it's important to understand the who, what, and how of each environment.

### Who is each environment targeted at?

#### *FileMaker*

Novice computer users, excel users, small business owners, small corporate workgroups, non-profit organizations and educators.

#### *Servoy*

ISVs (Independent Software Vendors), medium to large work groups within corporations, and independent software consultants who build custom applications for their customers.

## What are each environment's core competencies?

### *FileMaker*

FileMaker allows the creation and deployment of database applications – FileMaker is the non-SQL based, proprietary database engine where developed applications are created and served in a single proprietary physical disk file that contains the database schema; data; forms; and business logic.

With FileMaker, you can deploy the finished application to Mac OS X and Windows users (only) via a native client (FileMaker Pro must be installed on each client) and using “Instant Web Publishing” for up to 100 concurrent users via a web browser (FileMaker Server Advanced only); and to hand-held Windows Mobile devices using FileMaker Pro Mobile Edition.

The application server (FileMaker Server) can run on either Mac OS X or Windows – and an “Advanced” version is required to host ODBC and JDBC connections as well as up to 100 concurrent Web users.

Each FileMaker Server can service up to 250 concurrent users and up to 125 open database files at a time. Each FileMaker client can open a maximum of 50 database (physical files) at once.

### *Servoy*

Servoy allows the creation and deployment of “front end” applications to one or more SQL databases at the same time; as well as integration with existing legacy systems via web services; SOAP; .NET connectors; and Java.

Servoy also has the ability to deploy virtually the same application via a cross-platform native client (Servoy “Smart Client” via Java WebStart) on Mac OS X, Windows, Linux, and Solaris; and via a web browser on any platform that supports JavaScript - including Mac OS X, Windows, Linux, Solaris, Windows Mobile Devices, iPhone, Palm Treo, and others.

Servoy comes bundled with iAnywhere's SQL Anywhere, an enterprise-level SQL database at no extra charge – but ANY (or multiple) SQL databases can be used including: Sybase, Oracle, DB2, MS SQL Server, MySQL, Postgre SQL, OpenBase, FrontBase, and almost any other SQL database

with a JDBC driver.

The Servoy application server (“Servoy Server”) can run on any operating system platform including: Mac OS X, Windows, Linux, Unix, BSD, and Solaris; and is scalable based on hardware – from laptops to IBM Z Series (“mainframes”) computers.

There is no hard-coded limit to the number of concurrent users and no limitation on the number of tables or databases that a single application can use. The only limitations to speed and concurrency are based on computer hardware; network infrastructure; database setup and tuning; load on the network, etc.

## How do these products work from an overall architectural point of view?

FileMaker and Servoy take nearly opposite directions when it comes to overall architecture. FileMaker is based on proprietary technology based in C++ and Objective C; and Servoy is based on 100% open source Java. Below, we will compare and contrast the two environments to give you a basic overview of the product lines and what each product is used for.

### *FileMaker*

FileMaker is based on client-server technology (proprietary); includes an application builder/client (FileMaker Pro); an advanced application builder/client with more features (FileMaker Pro Advanced is targeted at “professional developers”); an optional basic application server (FileMaker Server) and a version of the server that allows ODBC/JDBC connections and web server “Instant Web Publishing” functionality (FileMaker Server Advanced). There is also an optional mobile client (FileMaker Mobile) for Windows Mobile devices (also runs on Windows CE). And FileMaker has a “custom web publishing” API based on XML and XSLT that allow developers to interact with their applications – but requires the FileMaker Advanced Server to function.

### *Servoy*

Servoy is also based on client-server technology (Java) and includes an application builder (Servoy Developer); a native client (Servoy Smart Client); a 100% HTML/CSS version for web browsers (Servoy Web Client); and an application server (Servoy Server). The Servoy Product Suite also includes an optional “Multi-Developer Server” that allows more than one developer to work on

the same project at the same time – and broadcasts all changes to all members of the development team; a “Headless Client” Java API (Servoy Headless Client) that allows developers to write their own JSP pages to interact with their applications and with other web services – and is included at no extra charge with the standard Servoy Server; and Servoy also has a “Disconnected Client” that runs on a disconnected computer – that allows synchronization back to one or more SQL database sources using the optional MobiLink software from iAnywhere.

## Looking at the “How” of Building Applications

### *Building the FileMaker Application*

Developers build FileMaker applications using FileMaker Pro or FileMaker Pro Advanced. A FileMaker application consists of a physical disk file on the developer’s local machine that can contain one or more database tables; one or more “layouts” (screen forms) that can be used for data entry or reports; zero or more “relationships” that represent bi-directional links between two or more database tables; and zero or more scripts that automate repetitive sequences (buttons, printing reports, etc).

FileMaker solution designers must define the database tables and the columns from within FileMaker Pro. They can then create a new “layout” with a wizard that allows them to choose the fields; set the graphic theme; optionally create summarization parts; and optionally create a script to view the current form in print preview mode.

As soon as the developer exits “layout mode” and enters “browse mode” – he/she can start using the application immediately – and can perform finds; create records; update records; delete records; etc., without any “programming” on the part of the developer.

The developer also has a few tools to show data from related tables – including looking up data from another table, auto-entering data into a column, and using a “portal” to display data from a related table. There are the standard line, box and oval object tools in addition to a button tool. Developers can place graphics on the background of a layout and turn them into buttons by linking them to scripts via a point-and-click interface.

The scripting environment in FileMaker is called ScriptMaker® - and consists of a modal dialog box that allows the user to string together various “Script Steps” of preset functionality in order to get desired functionality – much like a macro. These scripts are then saved within the same physical file as the data and the form definitions.

## *Building the Servoy Application*

Developers build Servoy applications using Servoy Developer and deploy them to the application server (Servoy Server). Users can then connect to the Servoy Server via Servoy Smart Client or Servoy Web Client to interact with the application. Like FileMaker, a Servoy application can contain data from multiple tables; can contain many forms used for data entry and/or reports; can contain zero or more “relations” - unidirectional joins between tables; and zero or more methods that automate repetitive tasks.

Unlike FileMaker Pro, Servoy is NOT a database. It does NOT store any of your data. All of your data resides in one (or more) SQL databases. Also unlike FileMaker, Servoy has no proprietary file format. Instead, all the information about your solution is stored as metadata in the form of rows and records in a SQL database repository (the repository can be any SQL database you choose).

Creating “forms” is very similar to creating “layouts” in FileMaker. The developer selects the database connection, then the table, and a blank form is created. The developer can then choose from a list of available fields in that table, and Servoy will automatically place the selected fields on the form along with their labels (optional). When the developer exits the “designer” mode – Servoy automatically binds the data to the SQL data columns and so the user can immediately perform finds, sorts, add records, delete records, edit records – all without any further programming on the part of the developer.

Like FileMaker, Servoy developers can also take advantage of a number of form and data elements – from “portals” to show related data; to buttons; background graphics as buttons; as well as line, box, circle, and polygon drawing tools; they can specify auto-enter data; and specify columns to contain data looked up from other tables.

However, Servoy Developer has additional tools which greatly reduce the complexity and maintenance issues found in a typical FileMaker solution. The ability for the Servoy developer to re-use their forms by placing them into “tabpanel” and then being able to programmatically control those tabpanels at runtime add a whole new dimension to the overall functionality and usefulness of Servoy applications.

Servoy also supports JavaBeans and Java Applets directly within the application. A JavaBean is a reusable software component that can be visually manipulated in builder tools, while a Java Applet is a small Java application that usually runs inside a browser. JavaBeans usually have some kind of user interface element to them: a slider control, a clock, a drag-and-drop pane, etc. A Java Applet is a small application that does something: monetary conversion, display molecule data in 3D, etc.

Servoy developers can use any existing JavaBean or Applet within their applications – and they can programmatically control them as if they were native Servoy objects. The Javabean/Java applet’s public properties show up in the native Servoy Properties Panel and the entire Java container object is exposed in the Servoy Editor - allowing the Servoy developer to attach listeners for specific events like clicks, mouse overs and more.

Servoy, unlike FileMaker, also supports events on objects. For example, field events include: `onFocusGained`, `onFocusLost`, `onDataChange`, `onAction`. Form-based events include: `onShow`, `onLoad`, `onRecordEditStart`, `onRecordEditEnd`, `onRecordSelection`, `onHide`; and also include events for all the menu options that a user can select: `onFind`, `onNextRecordCmd`, `onPreviousRecordCmd`, `onNewRecord`, `onDeleteRecord`, etc.

Having such a wide range of events at the Servoy developer’s disposal is a huge help when it comes to enforcing business rules, re-using code across the application, dynamically changing or highlighting data, showing/hiding/graying out objects, etc. These events allow Servoy developers the control they need to create adaptable, configurable, flexible applications.

## Deploying the Completed Applications - Client/Server

### *Deploying a FileMaker Application*

In order to deploy a FileMaker application, the FileMaker Pro (or FileMaker Pro Advanced) software must be installed on each client computer. This can be done via CD or network installation. Once the developer has finished creating the application and the client software has been installed, the developer would then upload the physical file to the server machine, login to a server administration application and then “start” the database file which makes it available to the other FileMaker Pro clients.

When deploying changes to an existing application, usually the FileMaker developer has to save a “clone” of the database (the structure only, without testing data), then take the existing production data “offline”, copy the production file to their hard drive, open FileMaker and import the data from each table from the production file into the file with the new changes. Once that process is completed, the developer can put the new file with

the current data back on to the FileMaker Server dedicated machine and re “open” it to make it available to all the users.

### *Deploying a Servoy Application*

The Servoy Client software also has to be installed on each client computer – but rather than installing from CD or a network install – the client has only to open a browser (pointed to the IP address of the Servoy Server) and click one button. The client software will download itself one time (about 3MB download), install itself, configure itself and then continue to launch. There is no need to install any JDBC drivers or configure any ODBC connections – because Servoy Smart Client only “talks” directly with the Servoy Server, and not directly to the backend database. If you want to update drivers or add databases or database connections, it only has to happen one place – on the server.

Not only that, but every time Servoy Client is launched (from the browser, a desktop shortcut or via a “deeplink” on your existing site or portal) – the software automatically checks with Servoy Server to see if an update is available. If there is an update, Servoy Client will download, install and

configure itself automatically and continue to launch. You NEVER have to manually update client software – ever – with Servoy.

Once the developer (or development team) has the application ready for deployment, it's a simple matter of exporting the solution out of the Servoy repository database (using Servoy Developer); uploading it to the application server (via a web page); and telling the newly-installed version to activate. You can roll forward to any new release or roll back to any previous release at any time. There is no limitation on the number of releases you can have – PER solution, PER Server.

Because Servoy is not a database and doesn't store data – there is no need for cumbersome and time-consuming data imports. Once the release is activated, clients who are already connected can be told (or forced) to close and re-open the solution to get the latest version; or continue to use the previous version until they open the application again – at which time the connected clients will see the new release.

The Servoy automated update technology saves hundreds of hours of lost productivity over having to take entire files (or sets of files) “offline” to do updates like you would have to do in the FileMaker environment. In addition, since most companies want to avoid downtime, it's very common to perform these updates after normal business hours and on weekends. This makes it very difficult to test the new or updated FileMaker application under load, to anticipate end user adoption of new features, often requiring more fixes and more downtime.

## Deploying the Completed Applications – Web

### *Deploying a FileMaker Web Application*

FileMaker offers two different ways to deploy your application to a web browser – the simplest way is called “Instant Web Publishing” (IWP). This is an automatic process that will render a web page that looks essentially like the application does in the FileMaker Pro client. There are similar navigation controls on the left-hand side for record navigation and searching for data, etc. All of the pages that are generated as part of IWP – are internal resources of the FileMaker Server and cannot be edited manually. IWP is limited to 100 concurrent web users per FileMaker Server.

You can also create an application that exports static HTML or XML pages to disk – but this is application-specific. You must write your own scripts and layouts and use calculations to tie everything together into the entire web page(s).

Finally, you can use “Custom Publishing” along with XML and XSLT to create completely custom web applications that access a FileMaker “database.” Unlike IWP, this tag-based language is unique to FileMaker, and it’s up to the developer to create and debug the pages themselves. The authors can trigger scripts to run; create, update, delete records; display and update information; perform finds, etc.

### *Deploying a Servoy Web Application*

Servoy also offers two different ways to deploy your application to the web. The first, and easiest way, is to use Servoy Web Client. There is no code required – you can direct end users to a launch page that shows all the solutions (like in an “Open” dialog box) by pointing to: `http://yourIP:8080/servoy-webclient`. Users can then just click on a solution and Servoy will automatically render the application in pure HTML and CSS, leveraging AJAX (Asynchronous JavaScript And XML) to dynamically update the web page without page refreshes – all without you, the Servoy developer, having to write any code.

You can also use the Servoy Headless Client API and JSP (Java Server Pages) to access your solution. This is really a great way to write custom web applications while sharing the same business logic as the Servoy Smart Client application. By including an eight-line wrapper, you can access your solution’s methods and data; and you can pass your methods parameters from the browser and receive back HTML, data, even Java objects - like recordsets, binary data and more.

Because the Servoy Headless Client API knows how to connect to your solution, you don’t have to specify a database connection; or a table name; or a SQL query. By writing your methods (scripts) in Servoy, you can leverage Servoy’s step debugger when developing your methods – and re-use existing business logic and validations.

Servoy also has built-in version control – and you can take advantage of this version control whether you’re developing an application that will be deployed via Servoy Smart Client or Servoy

Web Client. At any time during development, you can create a new “Release.” Once you deploy the release to Servoy Server (see below), you can choose to roll forward or backward to any other “Release” at any time, with or without clients attached – and you can do it from within a web browser.

## Maintaining and changing your application

### *Making Changes to a FileMaker Application*

If the FileMaker developer needs to make changes to the existing application they have two choices – they can either make changes (to the schema or layout changes/additions/deletions and/or script changes/additions/deletions or relationship changes/additions/deletions) to the live, production version while clients are connected or they must follow the same steps for “deploying” above: save a clone, “close” the file on the server, import the data, put the modified application back on the server and “open” the file.

### *Making Changes to a Servoy Application*

With Servoy, you simply export your solution out of the development repository - using Servoy Developer - then import the application into the production repository via a web browser. You can then activate the new release with or without clients connected; and you can force the clients to re-load the solution to get the new version, or simply continue using the older version. It’s up to you. The result is that you can control any Servoy application server from anywhere in the world – as long as you can make a HTTP connection to it.

## Extending Your Applications

### *Extending a FileMaker Application*

FileMaker has a plug-in API (“External Function Plug-ins”) that allows third-parties to create plug-ins that will enhance the capability of FileMaker beyond the default functionality that ships with the product. The FileMaker developer can only access these plug-ins from inside the calculation engine. The FileMaker plug-in API is based on the C programming language; and requires that the source code be compiled separately on both the Mac and Windows if you are planning a cross platform plug-in. This also means code changes if your plug-in is going to support user interface objects (open/file dialogs, system dialogs, pickers, etc.) that are platform-specific.

The External Function Plug-in API also only allows a single string to be passed in (or a comma-delimited list in version 7.0 and above) to the function, and only allows a single string to be returned from the function. Because of this single string return mechanism, many plug-in developers return a delimited list that the FileMaker developer must manually parse to retrieve multiple return values.

Another limitation is that the plug-ins are client-based – which means each end user has to install them in the “system” directory inside their individual FileMaker installation directory. It is possible to write a script that will pull down the plug-in from a FileMaker server – but again, this is up to the individual FileMaker developer to code; and it’s also up to the individual developer to come up with a plan to ensure that the plug-ins are the latest version, that they exist on the server, etc.

Because FileMaker’s External Function Plug-in API is only available to the calculation engine, it’s impossible to create a FileMaker plug-in that contains native objects that can be placed on a FileMaker layout. Everything must happen as part of a calculation – usually within a script.

### *Extending a Servvoy Application*

Servvoy also has a plug-in API that allows third-parties (and Servvoy itself) to create additional functionalities beyond the default functionality that ships with the product. However, Servvoy’s plug-in API is 100% Java- based, and goes all the way to the core of the product - having a dramatic effect on both how the code is written and maintained, as well as the type of functionality that is possible with the API. In addition to the plug-in API, Servvoy also natively supports JavaBeans and Java Applets – and allows the Servvoy developer to place them directly on a Servvoy form.

JavaBeans and Java Applets are small, pre-compiled Java applications that usually have to do with some user interface object (a slider control, chart, directory tree, etc.). There are hundreds of thousands of freeware, shareware and commercial JavaBeans and Java Applets available. In addition, all the public properties of a Javabean or Java applet show up in the Servvoy Properties Panel – as if it were a native object to Servvoy. So you can visually (and programmatically) set properties – as well as add your own Java “listeners” for clicks on the objects, and much more.

Because Servvoy's plug-in API 100% Java – you only have to write your plug-in code once (on any platform, using any tool) and the plug-in will work with any platform that Servvoy will run on (Mac OS X, Windows, Linux, Unix, Solaris) – without re-compiling for each specific platform. In addition, because each platform has its own JVM (Java Virtual Machine) – you can use the same code in your Servvoy plug-in to call an open/save dialog – and it's up to the JVM to make the OS-specific calls that draw the objects to the screen.

Unlike the FileMaker External Function Plug-in API, the Servvoy Plug-in API allows developers to pass in and receive back any valid Java object – including strings, numbers, binary data, arrays, recordsets and more. You can call your Servvoy plug-in from any method (script), or use them in calculations as well.

Also unlike the FileMaker plug-ins, the Servvoy plug-ins are server-based, not client-based. When you place a plug-in into the “plugins” folder inside your Servvoy installation folder on the server, that plug-in will be automatically downloaded to the client the next time the client starts up. It's the same process when you want to update (or remove) a plug-in. Simply put the new version (or remove it) from the plugins directory and that's it. Regardless of the number of clients, regardless of where the client is physically in the world, regardless of platform, regardless of whether you deploy via Smart Client or Web Client - the entire process is automatic with Servvoy.

## Integrating Your Applications

### *Integrating a FileMaker Application with Other Applications and Web Services*

FileMaker has a limited ability to interact with other databases (via ODBC natively or via JDBC with a commercial plug-in) and web services. Let's first look at SQL database integration. FileMaker has a native “Execute SQL” script step that will allow the FileMaker application to execute SQL statements (either written by the developer or as the result of a calculation) against a SQL database, and it will return a confirmation or error message.

The only built-in way FileMaker can communicate with a SQL source is via ODBC at the client level. This means that the appropriate ODBC driver must be installed on each local client computer, and the ODBC connection (DSN) must also be configured on each client that will communicate directly with the back end SQL database.

In the case of finding data for example using this simple SQL query:

```
select * from customers where cust_id = 101
```

The FileMaker script would return a recordset with a single record. The only way to view or use this information is to import the one record dataset into a FileMaker database. The same with multiple record recordsets – to view or use the information – recordsets must always be imported into FileMaker. As such, FileMaker is not an effective “front end” for SQL databases. In fact, it would be a nightmare trying to keep the data in FileMaker and the data in the SQL database(s) synchronized.

In order for FileMaker to communicate with outside applications or services, a special plug-in would have to be written - and the limitations of a single string output and use only in calculations would apply (see the description of the FileMaker plug-in API above). You can also send DDE Execute commands (Windows only), perform an AppleScript (Mac only), or send the operating system an event with parameters to do things like open a specific document.

With FileMaker 7 and above, it is possible to use a FileMaker database as an ODBC or JDBC data source. This means that it is possible to query a FileMaker database via ODBC or JDBC from another application. This is only possible if using the “FileMaker Server Advanced” product – and there have been numerous customer reports, blogs and reviews that say it is very slow. This type of interface is meant for simple queries with limited recordsets. It is not meant for large scale updates or selection of records.

### *Integrating a Servoy Application with Other Applications and Web Services*

Because Servoy is not a proprietary database (it doesn't store your data) – integration with multiple SQL databases is very easy. In fact, Servoy was made to create front-ends to SQL database data! There is nothing to “synchronize” because you're looking directly into the SQL database – without having to do any SQL coding yourself.

Servoy handles new records, updates, deletes, finds, omitting records, and more - automatically. If you can select fields in a dialog box, you can create a basic Servoy application that will talk to one or more SQL databases located anywhere in the world.

Not only that, you can actually cross-join data in multiple physical databases – even databases from different manufacturers. For example: Let's say you have a hosting service that hosts a Microsoft SQL Server database on Linux that contains order information; and you have a MySQL database on Mac OS X in your local office that contains customer information; and you want to build and deploy an application that shows all the customer data from MySQL along with all the invoice data from the MS SQL Server database on the same screen.

With Servoy, you can create and deploy that solution in less than 10 minutes without writing a single line of code.

Servoy also has a very active online forum (<http://forum.servoy.com>) and there is a free SOAP plug-in that will allow Servoy to talk to web services and other services via SOAP in very little code.

Of course, Servoy has the ability to communicate with the local client machine via AppleScript or via “Execute” commands just like FileMaker does – but with Servoy you also have the ability to include native JavaBeans and Java Applets that can provide a tremendous level of integration with legacy systems.

Because Servoy is not a database, you can always access your data from other applications via ODBC and JDBC without any performance limitations (other than those of the backend database itself) and without the fear of crashing Servoy Server.

Using Servoy's Headless Client API (SHC, as described earlier) you can expose your Servoy methods as web services for other web services and applications to use – so you really have a head start when you need to integrate with other SOA-based (Service Oriented Architecture) applications.

## Upgrading the Applications

### *Upgrading FileMaker*

From time to time, FileMaker changes the file format associated with their database (from FileMaker II to FileMaker Pro 1.0; FileMaker 2.x to 3.0, FileMaker 4,5,6 to 7.0). When that

happens – each physical database file (i.e. each application) has to be converted to the new file format. Depending on your application’s complexity and what features you use, the conversion process can be as simple as taking all the files off the existing FileMaker Server and opening them in FileMaker Pro (or FileMaker Pro Advanced) – which will do the conversion.

Other times, you may have to do the conversion, and then re-write parts of the application to function correctly with the new version of FileMaker. This can involve changing calculations, scripts, relationships, value lists, security settings, etc. - not to mention all the testing required for all the changes; and additional time to test the effects of any custom web deployments as well.

In addition, if there is a file format change – you need to upgrade all the clients and the server at the same time. The newer version of the FileMaker client is incompatible with the previous version server, and vice-versa. This creates a tremendous burden on personnel to update all FileMaker Pro client installations organization-wide – even for “casual” or occasional users of the system.

## *Upgrading Servoy*

Upgrading Servoy is very easy. Servoy Developer has a self-updating mechanism built-in. If there is a new version – you are prompted whether you want to download and update when launching the application. You can even choose to automatically download public beta versions.

Updating Servoy Server is easy – simply put the updater .jar file in the root of the Servoy folder and restart the application server (or service or daemon). That’s it. Servoy is completely self-configuring and previous versions of your solutions will continue to work.

## Licensing Methods and Costs

### *FileMaker Costs*

FileMaker is licensed per SEAT. This means that for every employee or computer in your organization, you need to purchase a copy of FileMaker Pro (\$299 or FileMaker Pro Advanced \$499) in addition to the FileMaker Server (\$999). If you want to connect to FileMaker via ODBC or JDBC or use the Instant Web Publishing – you must purchase FileMaker Server Advanced (\$2,499).

## Servoy Costs

Servoy is licensed per CONCURRENT USER, not per seat. This means that if you have 85 people in your organization, but only 50 will use the application at any one time, you only need to purchase 50 licenses. You can choose to deploy the Servoy Smart Client or Servoy Web Client - or mix and match - from the same Servoy Server at the same time. The cost per each concurrent client is \$289, and there is NO additional charge for Servoy Server – and NO additional charge for the Sybase iAnywhere SQL database engine that can be installed and used with your Servoy application.

Both Servoy and FileMaker offer discounts for purchases of multiple licenses at the same time. But, when you take into consideration the Total Cost of Ownership (TCO), Servoy is over 50% less expensive than FileMaker.

Suppose you have 85 people in your workgroup, and all of them will use your application – 50 of them full-time and the other 35 will only occasionally use it. Assume that you have never purchased FileMaker or Servoy before; that you have 2 developers; you want to have a staging server; and you also will need a “live” production server. The only server hardware box you have is one that you currently use for web testing – so you’ll need to buy a new production server (hardware) either way. Here’s how the costs would break down:

<i>Item</i>	<i>Servoy</i>	<i>FileMaker</i>
Client Licenses (60 for Servoy, 85 for FileMaker)*	\$13,740	\$20,845
Application Server License	\$0	\$2,499
1 Year Maintenance on Client Licenses	\$3,480	\$3,400
Developer Licenses (FileMaker Pro Advanced)	\$1,298	\$998
<b>Software License Total</b>	<b>\$18,518†</b>	<b>\$27,742†</b>
New Staging Server**	\$0	\$3,188
Additional Windows CALs for Staging Server Box (85)	\$0	\$4,949
New Production Server	\$3,188	\$3,188
Additional Windows CALs for Production Server Box (85)	\$0	\$4,949
Additional Linux CALs for Production Server Box	\$0	N/A#
<b>Hardware/OS License Total</b>	<b>\$3,188</b>	<b>\$16,274</b>
<b>GRAND TOTAL</b>	<b>\$21,706</b>	<b>\$44,016</b>

\*There are less Servoy licenses needed because Servoy is licensed per concurrent user, not per seat like FileMaker.

†All prices in US dollars. Prices based on published prices from both web sites at of March, 2007.

\*\*FileMaker recommends that FileMaker Server be the only process running on a server; while Servoy Server can share the existing web staging server. Staging Server and Production Server hardware prices are based on a Dell PowerEdge 2900 with (2) Dual Core Intel® Xeon® 5130, 4MB Cache, 2.00GHz, 1333MHz FSB, 2GB 533MHz (4x512MB), Single Ranked DIMMs, 36GB, SAS, 3.5-inch, 15K RPM Hard Drive, Microsoft Small Business Server 2003 Standard Edition – pricing as of April 2007.

#FileMaker Server requires Max OS X or Windows only – Linux/Unix/Solaris is not supported.

## Conclusion

By now it should be obvious that there are huge differences between FileMaker and Servoy. Although these products share some design-time and runtime paradigms (specify fields on forms, automatically handles data binding, finds, summaries, aggregations, record creation, updates, deletes, etc.) FileMaker and Servoy are architecturally completely opposite. FileMaker is a proprietary database, while Servoy is a user interface design and deployment platform that works with ANY SQL database(s).

It's important to choose the right tool for the right job – and it's equally important to have correct expectations on what each tool can deliver and how each tool can enable you to create, maintain and extend your own applications.

## About Servoy BV

Servoy BV is a privately-held company established to develop, sell and support the Servoy suite of products. The idea for Servoy started in 1998 by the four co-founders of the company -- being frustrated with the limitations of desktop database tools on one hand; and the complexity of web-based development tools, the steep learning curve, and the long development time on the other.

Today over 1500 companies and more than 10,000 developers are working with the Servoy suite of products. Companies like Symantec; Stanford University; Verizon; and UCLA hospital rely on Servoy for managing and presenting data to their customers and employees – providing rich applications over LAN, WAN and Internet connections. Servoy can count Apple, Oracle and Sybase among its technology partners.

The Servoy worldwide headquarters is located in The Netherlands (Amersfoort) where all research and development; as well as international sales and marketing activities are centralized and coordinated; Servoy Inc., Servoy's US office, is responsible for all US and North American sales and marketing events.

### International Headquarters

Servoy B.V.  
Algolweg 9A  
3821 BG Amersfoort  
The Netherlands  
Voice: +31 33 455 9877  
Fax: +31 84 883 2297

### Servoy USA

Servoy Inc.  
299 W. Hillcrest Drive  
Suite 115  
Thousand Oaks, CA 91360  
Voice: (805) 624-4959  
Fax: (805) 624-4958

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, AND SERVOY BV DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OR THE WARRANTY OF NON-INFRINGEMENT. IN NO EVENT SHALL SERVOY OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER INCLUDING DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, LOSS OF BUSINESS PROFITS, PUNITIVE OR SPECIAL DAMAGES, EVEN IF SERVOY OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY. SERVOY BV MAY MAKE CHANGES TO THIS DOCUMENT AT ANY TIME WITHOUT NOTICE. THIS DOCUMENT MAY BE OUT OF DATE AND SERVOY MAKES NO COMMITMENT TO UPDATE THIS INFORMATION.

©2007 Servoy, Inc. All rights reserved. Servoy is a trademark of Servoy, Inc., registered in the U.S. and other countries. All other trademarks are the property of their respective owners. Product specifications and availability subject to change without notice.