

## DAF technical profile

DAF (Database application framework) is a development platform developed and used internally by Ihsan Solusi to develop its applications. DAF is a complex development platform ranging from business component, data access, process and user-interface.

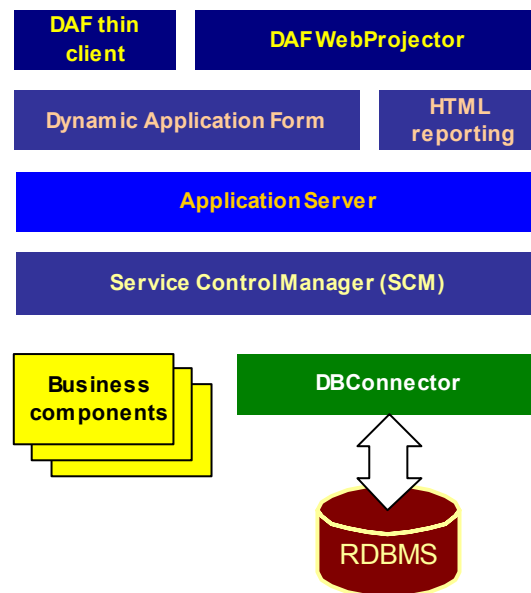
Using DAF, we develop various applications, including financial, medical and distance learning applications. The development of DAF was started in 2001 and still continues today.

DAF was developed under several fundamental paradigms:

1. Application should be built in a layered (or tiered) manner. User-interface layer must be separated from business rule layer, business rule layer must be separated from data access layer, etc.
2. Business rule development should be well-modularized.
3. The application development should be standardized, structured, and people independent.

### The elements of DAF

DAF is a big platform comprised of a set tools and libraries. The global scheme of DAF is described in the following diagram:



### **Business rule layer elements**

1. Business component platform

DAF provides a platform for defining business data semantics in an object-oriented manner. Data classes can be modularized in a class diagram and implemented in a component-based architecture. In the implementation of business data component DAF also provides a clear object-oriented syntax to data access. This object-oriented syntax helps programmers to quickly adapt with application semantics, improves efficiency and reduces errors caused by wrong SQL statements. Class code can be implemented using native code (for the sake of performance) or using Python language (for the sake of simplification).

## 2. DBConnector

DBConnector is a generalized database access interface for business components and application server. Using DBConnector helps the application and business component developed to be RDBMS-independent. DBConnector uses native, lightweight, unidirectional interface, hence improving access speed to the database. DBConnector is available for various popular RDBMS such as MS-SQL Server, Oracle, and Interbase.

## 3. Service control manager (SCM)

Service Control Manager is a library provided to connect to the business components and the database connector (DBConnector). SCM provides dynamic configuration of business components, integration with metadata, application-specific settings, etc. SCM also available as an ActiveX component, making it reusable by Microsoft-compatible tools such as Microsoft Office and Visual Basic.

### Process layer elements

#### 1. Dynamic application form

Dynamic application form is a language for describing user-interface forms. Most daily tasks of a database-application user-interface are available by default. Event handlers (both interaction or data events) could be defined using Python scripting language. Python is a type-relax, declaration-free and easy-to-use programming language.

#### 2. Reporting tools

Several reporting format is provided in DAF. There are text-based reports, graphical report, and HTML report. These reports are accessible and customizable by Python scripting facility.

#### 3. Application server

This is the heart of process layer. The DAFAppServer is a server that handles user application login / logout process, session management, delivering user interfaces (such as form and reports), task progress tracking, and processing data-related tasks including processing form results.

In the latest development of DAFAppServer, several advancements has been added such as stateful server, database connection caching and client-side resource caching. DAFAppServer now also supports multi-server configuration / clustering to serve more clients more responsively.

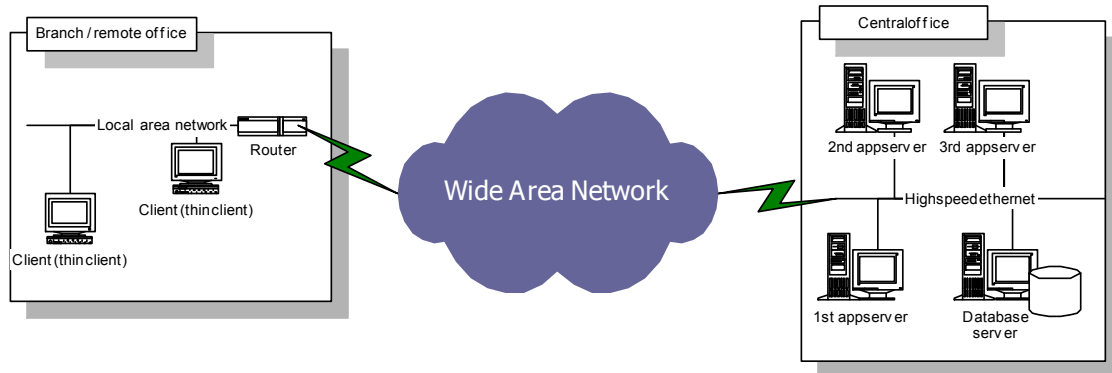


Figure 1 – DAF configuration with DAF ThinClient

## Presentation layer elements

### 1. ThinClient

The DAF ThinClient is a tiny foot-print Windows application (less than 2MB when installed, and less than 1.4MB of installation) required by a user to connect to applications in the DAF Application Server.

The DAF ThinClient requires almost no installation process (except unzipping) and can be accessed through folder-sharing in a local area network. No additional software is required. The communication between DAF ThinClient and the DAF Application Server requires only a narrow bandwidth, hence the configuration is available for wide-area-network environment.

### 2. WebProjector

The WebProjector is a web-based application that functions as a DAFThinClient to the DAF Application Server. The WebProjector provides a bridging function that allows web browsers to connect to applications in the DAF Application Server. This creates the "fourth-tier" in the DAF application environment. Using WebProjector also enables more configuration options in application deployment, including deploying application in the internet.

## The milestones of DAF history

The milestones of DAF history are listed in the following table:

Time	Version	Key features added
4 <sup>th</sup> quarter 2001	1.1 (initial release)	Object-oriented access and business object definition
1 <sup>st</sup> quarter 2002	2.0	Component-based business object architecture, dynamic link of runtime libraries, runtime scripting
2 <sup>nd</sup> quarter 2002	2.0i	CGI application support, user-interface component, business object implementation using script
3 <sup>rd</sup> quarter 2003	2.2i	Dynamic application form, application server, thin client
2 <sup>nd</sup> quarter 2004	2.3i	Improved application server, clustering, image and BLOB support
3 <sup>rd</sup> quarter 2004	2.4i (under construction)	WebProjector, performance improvement

# DAF enterprise configuration

