TRAVERSE®

TRAVERSE v11 Technology

White Paper
January, 2009

OPEN SYSTEMS INC.

TPLWPT

Contents

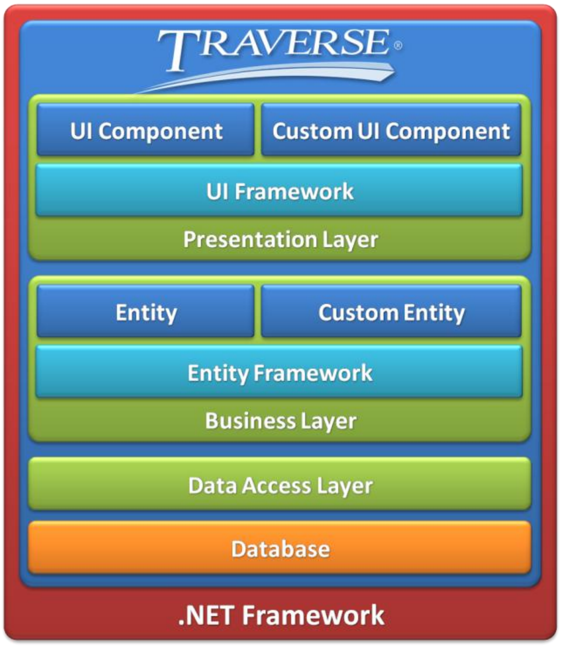
Information contained in this document refers to TRAVERSE software as it exists in January, 2009. The product design and functionality are subject to change, and there is no guarantee that a particular situation described in this document will not change in subsequent versions of TRAVERSE.

Summary

This document is designed to provide IT professionals with a basic outline of the technology and structure of the TRAVERSE version 11 software. The document authors presume that you possess an understanding of Microsoft tools and the operating environment. A References section at the end of this document provides URLs that you can use to further research specific topics relating to Microsoft's products.

All references in this document to TRAVERSE refer to TRAVERSE version 11 unless otherwise noted.

General Information



TRAVERSE is designed following object-oriented principles and is built on Microsoft's .NET Framework. The business processes can be deployed as software services in a service-oriented architecture (SOA.) The graphical user interface (GUI) used on workstations and mobile devices is written using Windows Forms and the C# programming language. The browser-based UI is authored using ASP.NET. The data is stored using Microsoft's SQL Server database. The supporting business logic may be found in either of two layers as deemed appropriate:

- Business objects, or entities, are assemblies written using the C# language.
- Transact SQL (T-SQL) code stored on the database server is used for data-intensive processing.

Reporting and data analysis for TRAVERSE are provided using a number of tools. The standard reports provided with the TRAVERSE software are designed using SQL Server Reporting Services (SSRS) and Microsoft Excel. Additional TRAVERSE applications that provide Excel-based financial reporting and data-mart analysis using OLAP technology are available as options. Note that SSRS is not a required component for viewing reports, but is required to change the report design. However, the Excel-based reporting is only available to users who have Excel installed.

The TRAVERSE menu system is designed to support the execution of additional reports created by users or third parties. Reports designed using Crystal Reports, documents using Microsoft Word's mail-merge capabilities, and many other reports can be easily integrated into the TRAVERSE menu structure.

Technology

While SQL Server remains the solid, flexible, and powerful database back end for TRAVERSE, we've chosen leading technologies such as C# as the basis for our program development. These cutting edge technologies allow us to provide you with a user experience and flexible software for a new era of doing business. An era in which software is easily adapted to your unique business needs and where you are in control of how and when you access information. You can be assured that these popular and powerful technologies will serve as a solid base for TRAVERSE well into the future.

Component	Language/Tool
Data storage	T-SQL / MS-SQL Server
Development environment	MS Visual Studio
Business layer **Data access layer**	C#
UI – Windows (workstation or mobile)	C# / Windows forms
UI – Browser	C# / ASP.NET
Primary reporting	SQL Server Reporting Services
Secondary reporting **(User reporting supported)**	MS Office (Word, Excel, Outlook) InfoPath/SharePoint Crystal Reports
Scripting	IronPython, JavaScript, others

Table 1 - Languages and tools used for TRAVERSE components

Server Technology

All editions of TRAVERSE use Microsoft's SQL Server for database management. The TRAVERSE media includes the free edition of SQL Server 2005, called SQL Express. This version of SQL Server is positioned as an entry-level SQL product for the small business environment. This edition of SQL Server has the following limitations that you should consider when you determine which edition to use for your TRAVERSE implementation:

- ➢ 4-gigabyte-per-database size limitation (In TRAVERSE, each company has a database that contains all of the application data for that company)

- Uses only 1 CPU, but can be installed on any server
- 1 GB addressable RAM
- No support for Analysis Services (data warehousing and business intelligence)
- Cannot serve as a replication publisher for transactional replication

To avoid these limitations or to accommodate larger user counts, a user-supplied Standard or Enterprise edition of SQL Server 2005 can be installed before implementing TRAVERSE. While it is possible to share a SQL database server among applications, we recommend a dedicated SQL Server (or SQL Server instance) for TRAVERSE.

The following table shows the various combinations of server and operating system products that can be incorporated in a TRAVERSE installation.

SQL Database Edition	Server Operating System
2005 Express **(Included with TRAVERSE)**	Windows Server 2008 (All editions) Windows Server 2003 (All editions) Windows XP (Professional) Windows Vista (Business or Ultimate)
2005 Standard or Enterprise (32-bit)	Windows Server 2008 (Standard, Enterprise or Datacenter) Windows Server 2003 (Standard, Enterprise or Datacenter) Windows Vista (Business, Enterprise or Ultimate)
2005 Standard or Enterprise (x64)	Windows Server 2008 (Standard, Enterprise or Datacenter) Windows Server 2003 (x64 Standard, Enterprise or Datacenter)
2005 Developer (32-bit)	Windows Server 2008 (Standard, Enterprise or Datacenter) Windows Server 2003 (Standard, Enterprise or Datacenter) Windows XP (Professional) Windows Vista (Business, Enterprise or Ultimate)
2005 Developer (x64)	Windows Server 2008 (Standard, Enterprise or Datacenter) Windows Server 2003 (x64 Standard, Enterprise or Datacenter)
2008 Editions	Details when commercially available

Table 2: Database Servers

Business Layer

The assemblies that constitute the business layer of TRAVERSE are written in C#. These assemblies can be deployed in a variety of ways, giving you the flexibility to determine the best configuration for your business. In the default configuration, these assemblies are delivered to each workstation to avoid the need for proxy classes. To simplify deployment and maintenance, the client installation for TRAVERSE uses ClickOnce technology so that updates provided to the environment are automatically delivered to the workstations.

When access to the business layer is extended outside the firewall, for wireless or Internet connectivity for example, it is exposed as a software service which provides secure access to the objects.

Client Technology

The user interface assemblies for client workstations and hand-held devices use Windows Forms and are written in C#. In the default configuration, these assemblies are installed (using ClickOnce technology) to the workstation or hand-held device. For a thin-client configuration, these assemblies are installed to a server running Terminal Services or Citrix.

Browser-based user interface components are built on ASP.NET and are installed on the Web server.

Structure

A typical TRAVERSE implementation includes a number of components:

- SQL Server – Database management. The TRAVERSE media includes the Express Edition of SQL Server. Alternatively, a user-supplied version of SQL Server Standard or Enterprise Editions can be installed.
- TRAVERSE database(s) – Company data and related programs. These databases are created during implementation or when new companies are being set up.
- Server Manager – TRAVERSE database administration tool. This tool is used to create, maintain, backup and restore company databases, manage security and permissions, and apply service packs.
- TRAVERSE Design Studio – A tool provided to allow for the custom design of forms, defining custom fields, scripting, data integration and other business specific requirements.

- .NET Runtime – Client GUI management. If not already present, the .NET v2.0 runtime component is installed to each workstation.
- Client application programs – Windows Forms assemblies. Simple functions, such as Terms Code and Distribution Codes maintenance, are aggregated into single assemblies while more complex maintenance and transaction functions have their own assemblies.
- Business layer components – Compiled C# assemblies. There is generally one business layer component for each business object, such as Customers, Items, etc.
- Productivity Report Files – Excel spreadsheets. These spreadsheets can be stored as local files customizable be each user, or as shared files on a server. An Excel plug-in is installed that provides security validation to the TRAVERSE data.
- Client installation components – Used to install the client workstation components using ClickOnce.

When the Typical (default) option of the installation wizard is used, the components are installed as shown in the following figure.

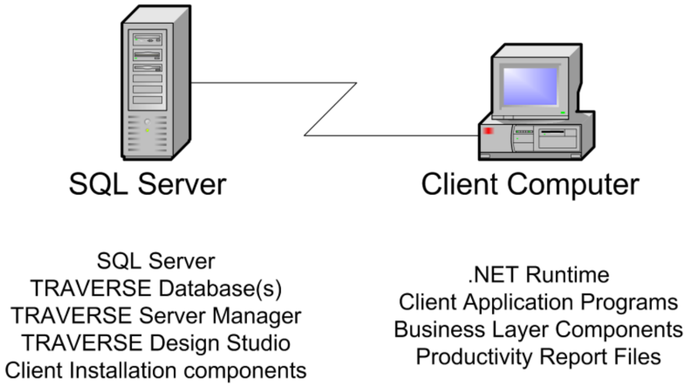


Figure 1 - Typical (default) TRAVERSE installation

You can configure a TRAVERSE implementation in a different manner to better suit your business. Most of the components are portable and you can place them on any server that exists within the network environment. Note that having the business layer assemblies on client and server avoids the need to generate proxy classes.

TRAVERSE can also make use of the flexibility of SQL Server. It is possible to use server groups and to distribute log files to specific servers, which can be helpful in a transactional replication environment.

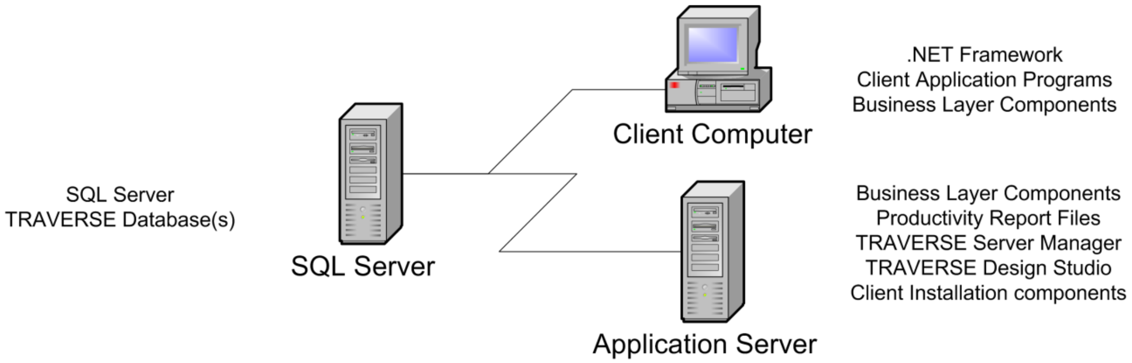

.NET Framework
Client Application Programs
Business Layer Components
Client Computer
SQL Server
TRAVERSE Database(s)
SQL Server
Business Layer Components
Productivity Report Files
TRAVERSE Server Manager
TRAVERSE Design Studio
Client Installation components
Application Server

Figure 2 – One example of a distributed TRAVERSE installation

Ultra-Thin Client Considerations

TRAVERSE can also be implemented using an ultra-thin client configuration such as Microsoft's Windows Terminal Server.

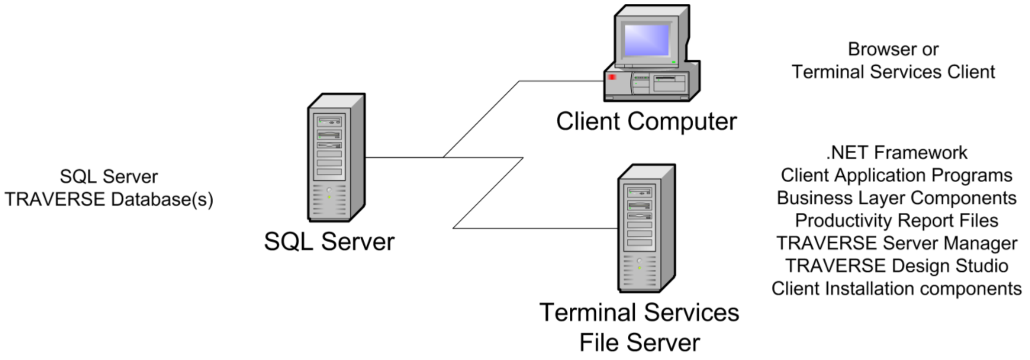


Figure 3 - Ultra thin-client TRAVERSE installation

Internet and Wireless Deployments

TRAVERSE can use the public Internet and wireless infrastructure to extend the software's reach beyond the firewall. This is accomplished using software services to ensure secure access to the desired functionality.

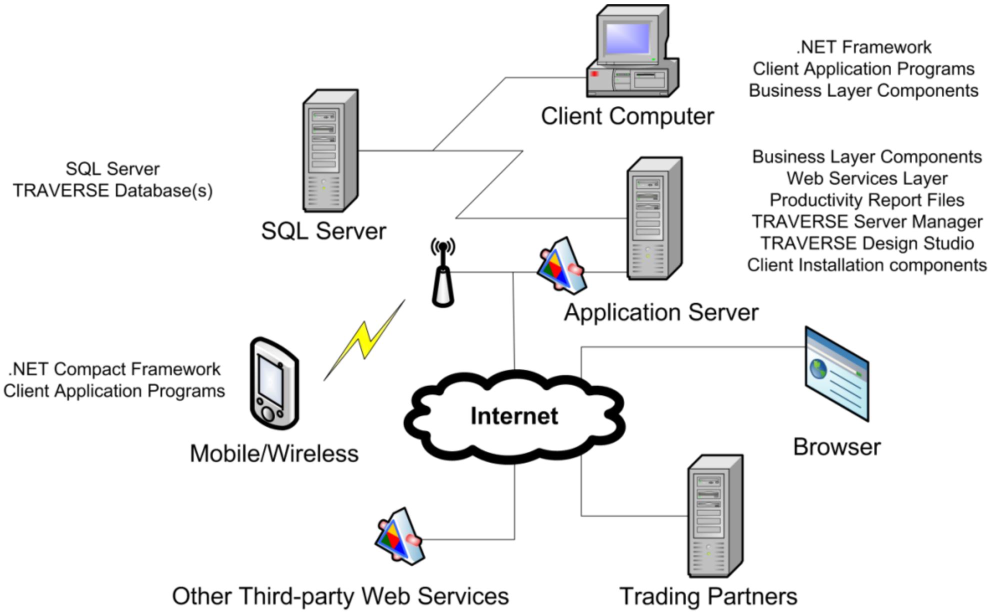


Figure 4 - Internet and wireless deployments

Database Integrity

Data integrity at the server level is maintained using several facilities available in the SQL Server RDMS. Proper data types are defined for data fields to enforce different types of acceptable data; specialized stored-procedures called triggers are used to maintain data relationships, and primary-key and unique index constraints are used to enforce non-duplicable values. All create, read, update, and delete (CRUD) operations are wrapped in database transactions as atomic units of work.

Where complex database operations are being performed, such as in a batch posting stored procedure, SQL transactions are used to ensure that all of the complimentary database updates required to maintain a sound accounting system are performed. If any problems occur during any of the attempted database updates within a transaction, the database status is rolled back to its condition before any of the transaction updates were attempted. This provides an opportunity to discover and repair the problem within the database before executing the function again.

Security

TRAVERSE maintains a catalog of users, including domain users, with access to specific application functions based on role memberships. User and Role are unique entities within the scope of the TRAVERSE instance. Roles are assigned to the company databases to which they should have access. Roles can be granted or denied permission to functions at the company level. Users are assigned to one or more roles and inherit all the permissions of roles.

TRAVERSE creates a single login on the SQL Server instance hosting the databases and grants access to only the databases required for TRAVERSE. This login is used by TRAVERSE to impersonate the users once the security credentials have been validated by the License Server. Since this login should have database owner or higher privilege on SQL Server, it is encrypted.

Security setup is managed using the TRAVERSE Server Manager application.

Adapting to your business

Even as the functionality and sophistication of business management software continues to grow, there can be parts of your business process that don’t map well onto that functionality. To be most useful, a software solution must be configurable to adapt to your business without expensive customizations.

You can adapt TRAVERSE to meet those needs through any combination of four methods: personalization, design, integration, and customization.

Personalize

Design

Integrate

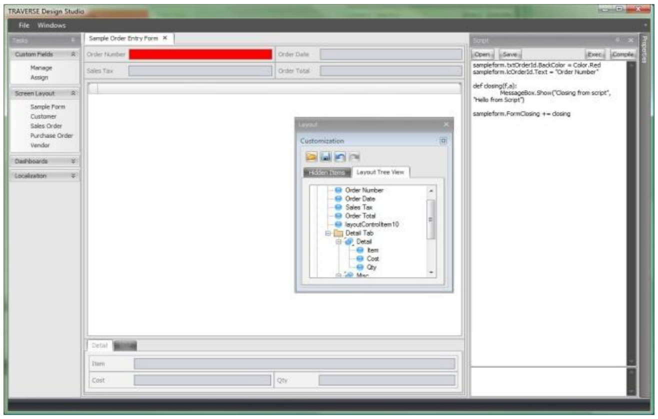
Customize

Personalize

Individual users can personalize the software by adjusting the menu terms and organization, changing the basic form navigation, and by creating and saving individualized ‘views’ of data for inquiry and reporting.

Design

Companies can create user- or role-based variations of the user-interface, add custom fields, manage workflow and more using the TRAVERSE Design Studio. The Design Studio also supports the use of scripting languages to add validations and process branches to the software.



```
sampleform.txtOrderId.BackColor = Color.Red
sampleform.lcOrderId.Text = "Order Number"

def closing(f,a):
        MessageBox.Show("Closing from script",
"Hello from Script")

sampleform.FormClosing += closing
```

Figure 5 - TRAVERSE Design Studio

Multiple variations of a given function can be saved and assigned to specific users or roles. This provides an easy mechanism to control which users have access to specific information. For instance, a data entry person may not be allowed to see the cost information on a quote or order, but the sales manager working with the same data entry form can see that information.

Adding custom fields to the software requires only two steps using the Design Studio. First, you define the field, giving it a name, screen prompt, data type, any field validations, etc. Then, you associate that field with a TRAVERSE object, such as an inventory item, a customer, or a sales order. That field is added to the function used to maintain that object, and automatically becomes available for inclusion on any of the personalized reports or inquiries based on that object. Note that all custom field definitions and values for a given object are stored as a single XML data element in the primary table for that object. This design avoids metadata changes to the schema while retaining the ability to easily access the data with a simple SQL XML query.

A key aspect of the Design Studio is that any design changes that are done with it are preserved when service packs and other updates are applied to the software.

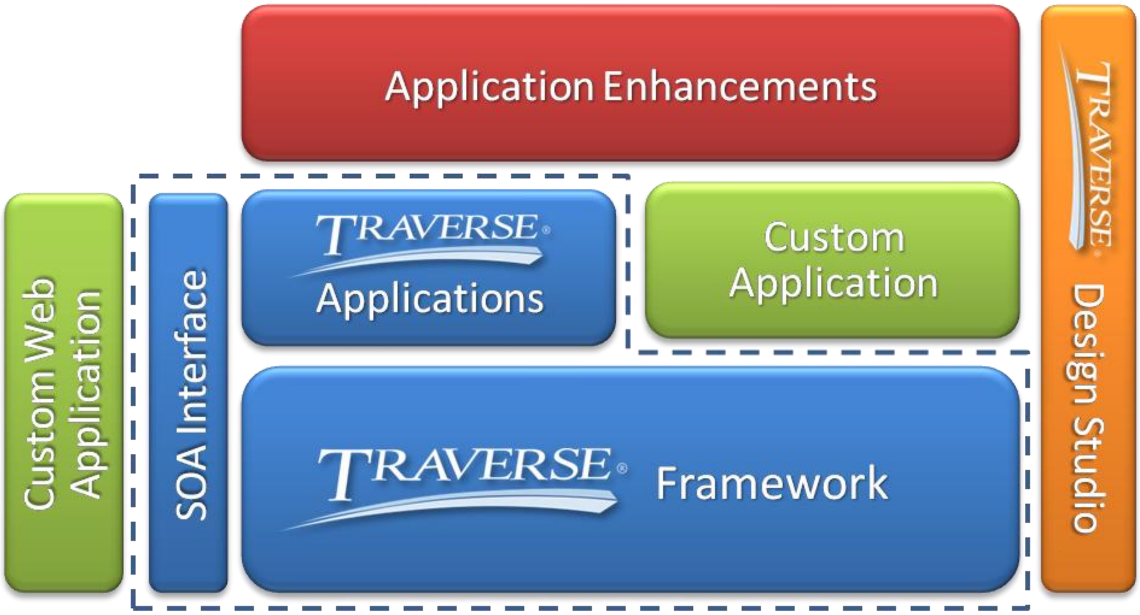
Integrate

The TRAVERSE Design Studio also includes the TRAVERSE Integration Engine (TIE), a set of data mapping and scheduling tools that can be used to integrate your Web store, trading partner, or other proprietary or third-party applications to TRAVERSE. Using the integration tools in the Design Studio ensures that data coming into your system is passed through the same validations and business processes as manually entered data.

Customize

The TRAVERSE applications are built on a complete framework designed to simplify the extension of functionality necessary to meet the specific needs of a company. A detailed diagram of the framework is shown in the appendix.

Companies that want to build in-house solutions to meet specific business needs can build application extensions or custom applications using this framework.



Any application extension or custom application built using the TRAVERSE framework automatically inherits the personalization and design points outlined above.

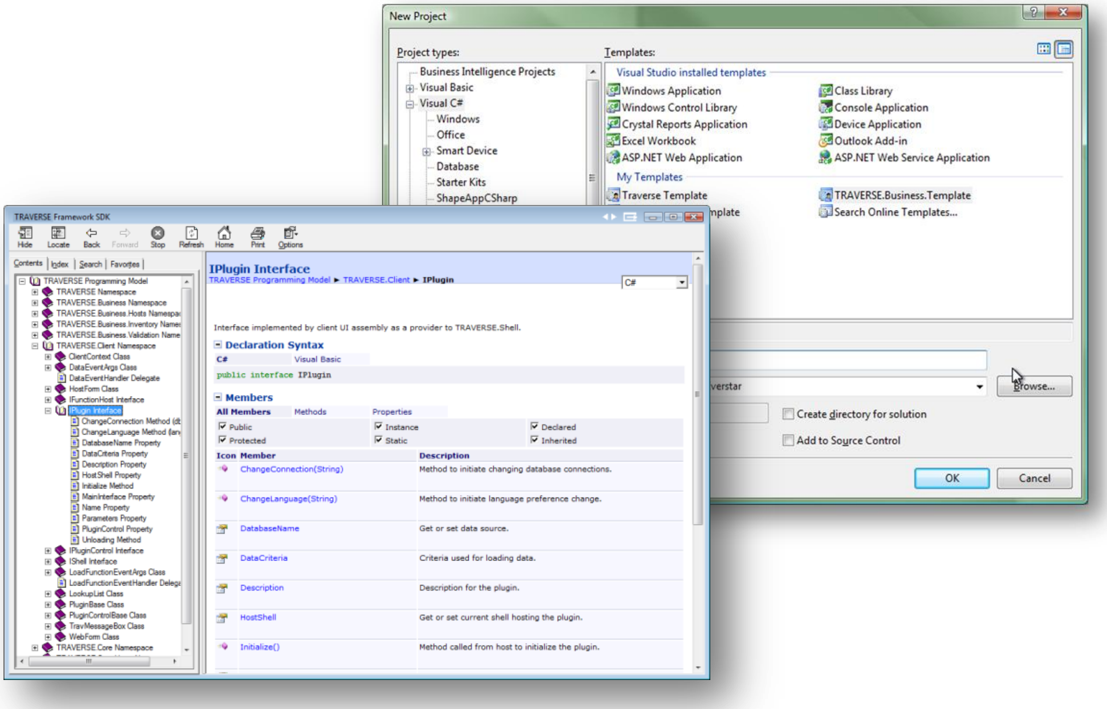
Using object-oriented techniques, such as class inheritance, ensures that your customizations will generally continue to operate after the application of service packs and other updates.

Software Developer's Kit (SDK)

The TRAVERSE Software Developer's Kit is an invaluable tool for anyone building applications or application enhancements for use with TRAVERSE. The SDK provides documentation and tools that can be used to build TRAVERSE plug-ins that look and function just like the standard TRAVERSE software.

Some of the primary components included in the SDK:

Standards Guide – A complete guide to the naming conventions, form and report standards, coding conventions, online help, and testing standards used.
Database schema – Electronic listing of table and column definitions.
Full object definition – Complete documentation of all framework classes and methods.
Visual Studio plug-in – Project templates to create TRAVERSE compliant projects.
Code generator – Generates TRAVERSE compliant code based on custom table schemas.



IPlugin Interface
TRAVERSE Programming Model ► TRAVERSE.Client ► IPlugin

Interface implemented by client UI assembly as a provider to TRAVERSE.Shell.

Declaration Syntax

C# | Visual Basic

```
public interface IPlugin
```

Members

All Members | Methods | Properties

☑ Public ☑ Instance ☑ Declared
☑ Protected ☑ Static ☑ Inherited

Icon	Member	Description
	ChangeConnection(String)	Method to initiate changing database connections.
	ChangeLanguage(String)	Method to initiate language preference change.
	DatabaseName	Get or set data source.
	DataCriteria	Criteria used for loading data.
	Description	Description for the plugin.
	HostShell	Get or set current shell hosting the plugin.
	Initialize()	Method called from host to initialize the plugin.

Figure 6 - Class definitions and Visual Studio plug-in from the TRAVERSE SDK

References

For More Information

On the Internet:

Open Systems, Inc.
http://www.osas.com

TRAVERSE
http://www.osas.com/TRAVERSE

Microsoft .NET Framework
http://www.microsoft.com/net
http://msdn.microsoft.com/en-us/beginner/bb308799.aspx

Microsoft SQL Server
http://www.microsoft.com/sql
http://www.sqlpass.org

Object Oriented Programming (OOP)
http://en.wikipedia.org/wiki/Object-oriented
http://msdn.microsoft.com/en-us/beginner/bb308750.aspx

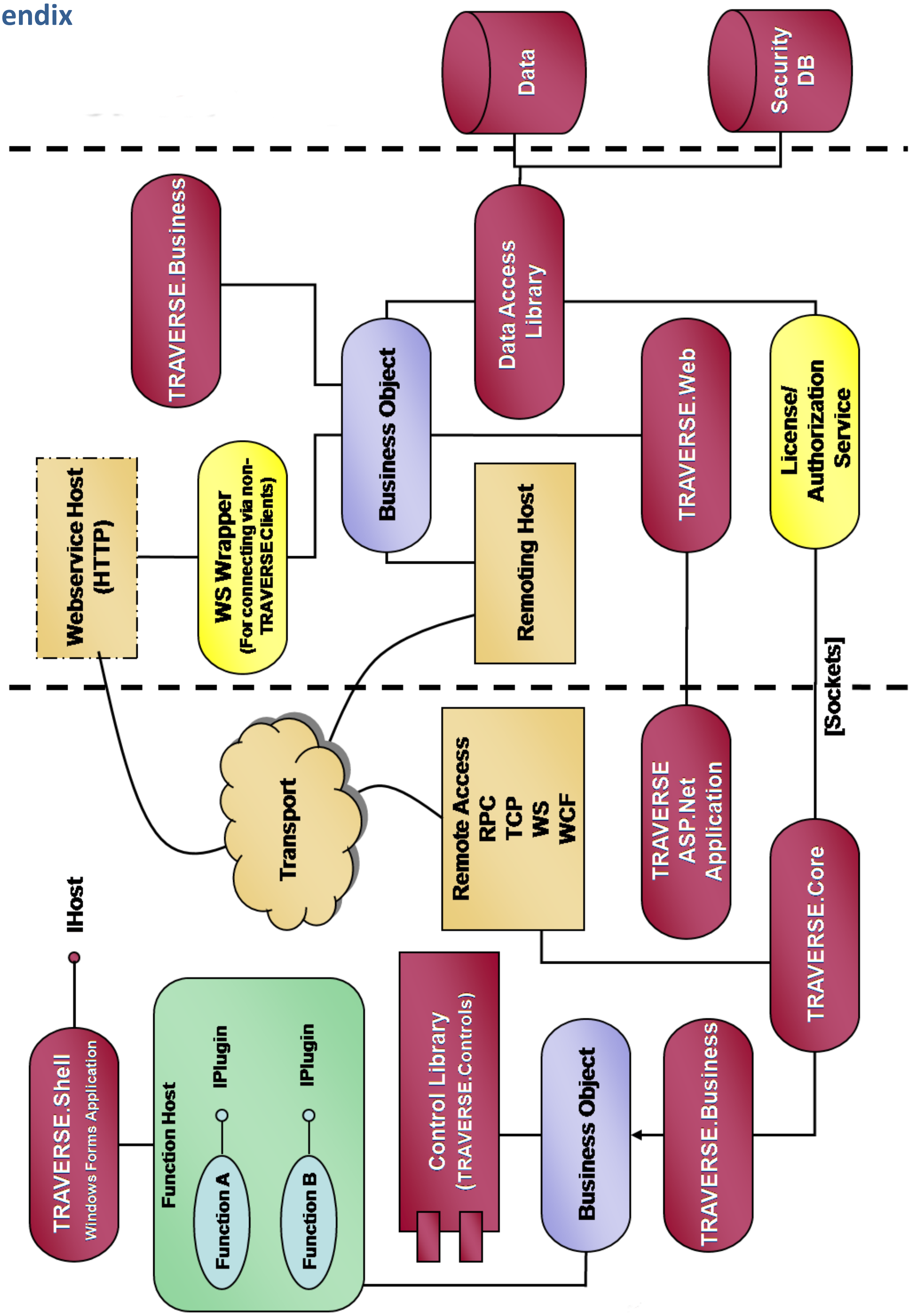
Service Oriented Architecture (SOA)
http://en.wikipedia.org/wiki/Service-oriented_architecture
http://msdn.microsoft.com/en-us/architecture/aa948857.aspx

Books:

“Microsoft Visual C# 2005 Step-by-Step (2005 Edition)” - John Sharp
“Applied Microsoft .NET Framework Programming” - Jeffrey Richter

Appendix

TRAVERSE Framework Diagram


TRAVERSE.Shell
Windows Forms Application
IHost
Function Host
Function A
IPlugin
Function B
IPlugin
Control Library
(TRAVERSE.Controls)
Business Object
TRAVERSE.Business
TRAVERSE.Core
Transport
Remote Access
RPC
TCP
WS
WCF
TRAVERSE
ASP.Net
Application
[Sockets]
Webservice Host
(HTTP)
WS Wrapper
(For connecting via non-
TRAVERSE Clients)
Remoting Host
Business Object
TRAVERSE.Web
License/
Authorization
Service
TRAVERSE.Business
Data Access
Library
Data
Security
DB