

**Visible Business Templates™  
An Introduction**

**By Graham Sword  
Principal, Consulting Services**

*This document provides an introductory description of Visible Business Templates™ — an overview of their construct and possible uses within a modeling effort. A detailed demonstration may be obtained by contacting Visible sales and marketing 800-6Visible (sales@visible.com).*

## Introduction

The concept of component-based application development dates to the structured development revolution of the 1970's. At that time, developers attempted to maximize code reuse via reusable models. However, the structured paradigm does not provide a mechanism for building entire systems from pre-built components nor does it effectively describe a method for reusing models early in the life cycle.

Object oriented design and development techniques then emerged with promises of reuse. Generally, however, the greatest reuse is still achieved with the use of the software components developed later in the life cycle.

Every development methodology comes with a paradigm invisibly attached. The assumptions and expectations of its practitioners are largely a product of familiarity with specific tools and techniques. On Object-Oriented analyst, for example, never leaves home without an extensive class library. And the use of a class library encourages a view of software development as “design by exception” – Reuse by default.

The conventional data modeler approaches problems with an entirely different paradigm. Typically this results in the “blank slate” approach – entities are modeled one at a time, starting from scratch. Every model is assumed to be unique. With experience, lots of it, the data modeler gradually comes to a deeper realization. Certain patterns begin to look familiar, regardless of organization or industry. Almost subconsciously, a collection of reusable constructs is amassed. Then, with little apparent effort, the experienced analyst “comes up” with business model construct that amazes novices. Effectively the expert data modeler has changed paradigms and is applying an object-oriented approach to data modeling. Having pattern awareness is like being armed with an object class library.

*Visible Business Templates* attempt to fulfill the ideal of total model reuse through familiar objects and patterns. The key to this approach is to consider a model as constructed of reusable components; i.e., object classes. Every model contains fairly common object classes developed over and over again in different guises. By raising the level of abstraction we are able to generalize about the object classes and their relationships within a single model — thus, the *Visible Business Templates*.

*Visible Business Templates* also add an additional frame of reference. Each object class may have one or more levels of abstraction based upon the business perspective from which the object class is viewed. For example, an entity viewed from a sales perspective will have different characteristics (attributes and associations) than the same entity viewed from a manufacturing perspective. Every *Visible Business Template* is constructed with these different perspectives in mind.

## Scope and Purpose

*Visible Systems Corporation's* comprehensive *Visible Business Templates* provide the basic data architecture needed to build integrated information systems.

The premise of *Visible Business Templates* is that:

1. all businesses, even diverse businesses perform a set of common functions, e.g. Marketing, Sales, Research, Production, Purchasing, Human Resources, etc.,

2. each business function performs many of the same activities, e.g. Contract Management, Contact Management, Document Management, Inventory Control, Budgeting, Accounting, Hiring, Training, etc.,
3. many of the activities act on similar data, e.g. organizations, people, skills, products, services, facilities, locations, assets, liabilities, accounts, documents, etc.,
4. in many cases the similar data is in fact the same data, e.g. Mr. Jones, Ms. Smith, Washington DC, Dallas, etc.,

In the *Visible Business Templates*, closely related data are grouped by subject area (business objects). While many business activities can be supported by data from a single subject area, other activities need data from several subject areas. The entire set of *Visible Business Templates* show the interdependency of the subject areas. By grouping the activities performed by business function; the subject areas, hence the data, needed to support the business function are determined.

*Visible Business Templates* are the result of years of experience gained by *Visible* consultants in many diverse businesses. In addition, the model provides the basic data relationships that often are not stated during the “system requirements gathering” phase of major projects.

*Visible Business Templates* contain common object classes that support multiple activities; rather than activity specific object classes often developed over and over again in different guises. Recognition of repeating business patterns during the development of business models over the years has permitted *Visible* to establish these object classes.

### **The *Visible Business Template Construct***

*Visible Business Templates* are a set of high-level models that contain nearly fifty business subject areas or object classes and encompassing over 600 entities with 1,000 attributes. Each of the object classes is centered on a “kernel” entity. The kernel entities were established based on their relevance in the business environment. The object classes constructed around the kernel entity vary in complexity. The object class PERIOD, for example, consists of only 3 entities. Others, such as PERFORMANCE are very complex and consist of between 20 and 30 different entities.

The following table presents a list of some of the object classes used in the construction of the *Visible Business Templates*.

➤ ACTIVITIES	➤ DELIVERY COMMITMENTS	➤ PERFORMANCE
➤ APPLICATION FOR FUNDS	➤ DOCUMENTS	➤ PERIODS (TIME)
➤ APPLICATION FUNDING	➤ FACILITIES	➤ PERSONS
➤ APPLICATION OBJECTS	➤ FINANCIAL ACCOUNTS	➤ PHYSICAL ASSETS
➤ APPLICATION ROUTING-TRACKING	➤ FINANCIAL ASSETS	➤ PRICES
➤ ASSETS	➤ FINANCIAL CENTERS	➤ PROCESSES
➤ BUSINESS NEEDS	➤ FINANCIAL TRANSACTIONS	➤ PRODUCTS
➤ BUSINESS PARTIES	➤ INTERNAL POSITIONS	➤ PROJECTS
➤ BUSINESS PARTY RELATIONSHIPS	➤ LAWSUITS	➤ QUALIFICATIONS
➤ BUSINESS PLANS	➤ LOAN APPLICATIONS	➤ RESOURCE REQUIREMENTS
➤ BUSINESS SOLUTIONS	➤ LOCATIONS	➤ RESOURCES
➤ CHECKING ACCOUNTS	➤ MARKET OFFERINGS	➤ SERVICES
➤ CONTACTS	➤ MARKET SEGMENT CRITERIA	➤ SPECIFICATIONS
➤ CONTRACTS	➤ MARKET SEGMENTS	➤ STANDARDS
➤ DATA OBJECTS	➤ MEASUREMENT UNITS	➤ STATEMENT ACCOUNTS
➤ DISPUTES	➤ MODEL OBJECTS	➤ SUBJECTS REFERENCED
	➤ ORGANIZATIONS	➤ WORK

### Object Classes and Properties

Each object class or entity, and each object class property (e.g. closely coupled entity or attribute) is not only associated with a complete definition; but includes the reason or purpose for it's existence.

### Model Views

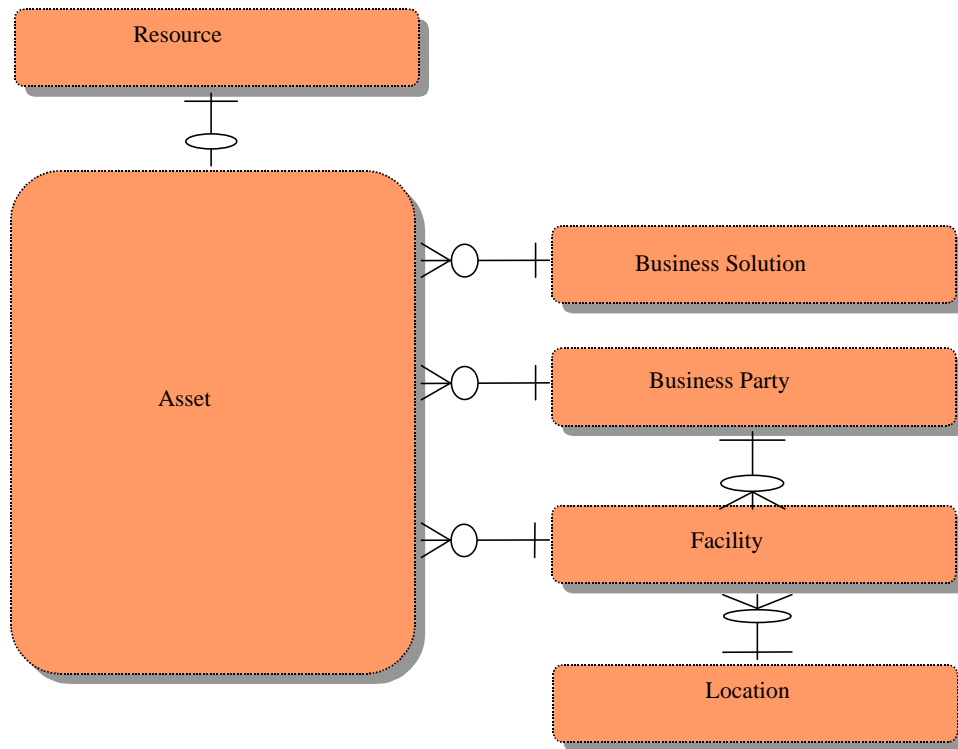
Together, *Visible Business Templates* provide a single integrated model that is presented in a number of views to facilitate understanding. *Visible Business Templates* have two levels of views — the object-view and object-class-views named for each of the kernel entities.

The object-view provides a big picture perspective of how the object classes relate to one another.

An object class view consists of a number of entities that are closely coupled to the “kernel” entity. These closely coupled entities are viewed as properties of the “kernel” entity. The object class may also reference entities that are a part of another object class view. Entities “belonging” to other object class views (properties of other object classes) are located within the object class view without view authorities to aid in visualization of object class relationships. For example, the entity **LOCATION** is referenced by a number of other object class and is, therefore, visible within each of the other referenced object class views.

**Object View**

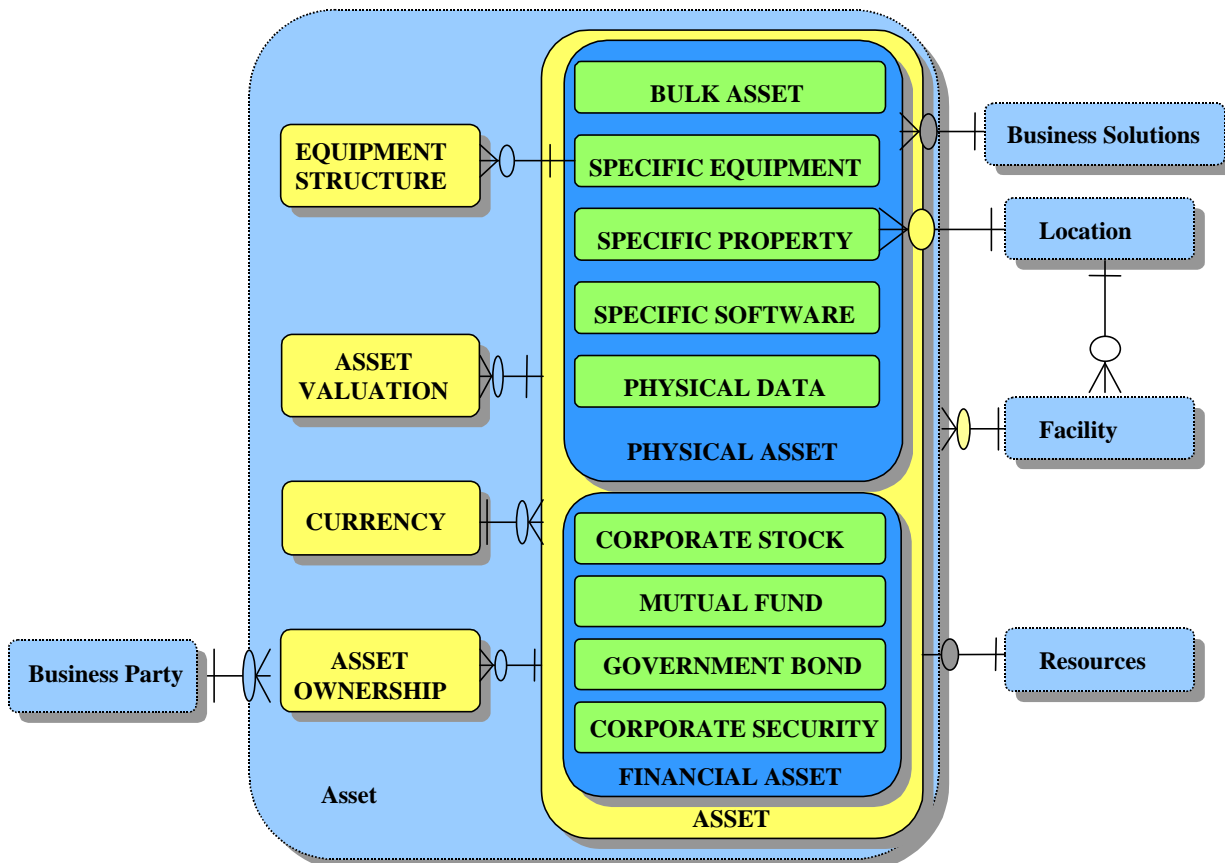
Each object class is named after a central entity called a “kernel”. The data model constructs for each object class vary greatly in complexity; they range from a few entities to twenty or thirty entities.



This diagram illustrates six of the object classes and their relationship to one another.

### Object Class View

The following section provides a brief description of an example object class with an accompanying diagram. The diagram shows the entities that comprise the object class, entity relationships within the object class, and relationships between object classes.



### ASSET

**Definition:**

An asset is anything of value, whether tangible (e.g., equipment, materials) or intangible (e.g., royalty agreements, patents.) that is owned by a person, association, corporation or estate.

Assets refer to SPECIFIC items. For example, "Ford Explorer" defines the characteristics of an asset; while "Ford Explorer with the Vehicle Identification Number: 1AB1234CD56789EF" owned by John Doe is an asset of John Doe.

**Purpose:**

Identifies the entire property of an individual, association, corporation or estate applicable or subject to the payment of his or its debts.

Identifies the items on a balance sheet of an organization, association, person or estate (identified in the model as a Financial Center) showing the book value of property owned.

An asset is a resource at the disposal of the owning entity to use in performing its function.

The location of the asset provides information on its availability.

An asset is classified, as a potential business solution to a business need.

**Using the *Visible Business Templates***

*Visible Business Templates:*

1. Provide the basis for reengineering the enterprise
2. Promote reuse of business objects, comprised of both data and activities, across the various functions of the enterprise.
3. Establish direction for system development,
4. Significantly increase the likelihood that new systems will effectively meet business needs
5. Significantly reduce system development time
6. Significantly reduce system maintenance

Using *Visible Business Templates*, an organization need only add the model components that differentiate them from their competitors before system development can begin. This cuts significant time from major development projects while assuring the developed system is part of an integrated solution and does not become a new legacy system. Using the *Visible Business Templates* as a starting point, our clients leverage the many years of experience *Visible* has in developing integrated information solutions.

*Visible Business Templates* can be considered a tool kit of familiar objects and patterns of relationships between objects and object classes. The first step in using the *Visible Business Templates* is to become familiar with the object classes. The next step is to evaluate the relationships between the object classes.

In a modeling session, the emphasis will change from one of discovering one object at a time to:

- Kernel discovery;
- Kernel extension; and

➤ Kernel combination.

Kernel discovery is recognizing references to universal classes under different names. For example, organizations have Employees, Managers, Customers, etc. These entities can all be represented by the object-class “PERSON”. The context the “PERSON” holds in an “ORGANIZATION” is represented by the relationship.

The challenge is then to evaluate the objects identified, during the discovery effort, against each other and the entities already established in the *Visible Business Templates*. In many cases, this is a matter of establishing the most meaningful level of abstraction. As new kernel entities are discovered, their relationship(s) to existing object classes are determined.

Use the kernels as building blocks, adding detail as required by the analysis of the business. The detail added to the objects (kernel extension) can be in the form of additional entities, attributes, or relationships.

In some instances, *Visible Business Template* object classes may be too detailed and require that one or more object classes be combined. As a rule, we expect most changes to involve additions to the model rather than modifications to the existing constructs.

In most cases, we assume that the changes will be to the existing objects within the model. For example, each entity in the model has been populated with attributes with suggested domains. Most frequent changes to objects will, therefore, include:

- Changes to static lists, especially in static lists that drive out secondary kernel entities;
- Renaming of entities or attributes with corresponding refinements to the definitions;
- Functional substitutions such as specifications of domains; and
- Population of rule entities.

A model assembled for a specific business from universal object classes should still be a completely customized solution. It will be more stable than a model built from scratch because it is built from pre-tested components but it will provide the business with a unique identity.

For organizations that have already developed one or more data models or perhaps have begun building an enterprise data architecture, *Visible Business Templates* can serve as a quality benchmark against which existing models can be compared. Existing models can also be integrated with *Visible Business Templates* to achieve data standardization and to ensure information system interoperability.



---

*Visible Solutions, Visible Success Stories, Visible Viewpoints, and other Visible white papers are available from our web site (<http://www.visible.com>).*

For more information concerning *Visible Business Templates* please contact:

**Alan Perkins**  
**Vice President, Client Services**  
**(703) 739-2242 ext. 225**  
**[alan.perkins@visible.com](mailto:alan.perkins@visible.com)**

### **About Visible**

*Visible Systems Corporation (Visible) is a world leader in software engineering solutions, tools, training, and consulting services. Founded in 1981, Visible has consistently pioneered products and practices for efficiently and effectively engineering mission-critical systems.*

*Visible's modeling and design tools are designed for software engineers, analysts, and designers interested in taking an architected approach to developing quality software solutions quickly.*

*Visible brings much greater value to customers by providing not only exceptional, proven software development products, but also skills improvement training, and on-site consulting. Visible helps software developers successfully support the changing needs of the enterprise, and helps the enterprise fully leverage its investment in information systems and technology.*

*Visible, a member of the Software Productivity Consortium, was founded to improve the effectiveness and efficiency of software development worldwide. Visible helps clients assimilate a rigorous and repeatable, disciplined, tool-supported Enterprise Engineering methodology for software component engineering through training, coaching, and mentoring.*

- *Staff skills and productivity improve so that better software is developed faster.*
- *Standards are established and enforced.*
- *Metrics are identified and utilized.*
- *Appropriate tool sets are acquired and used.*
- *Software development quality becomes consistent and repeatable.*
- *Changing requirements are met rapidly and well.*
- *Maintenance costs (typically, 70% to 80% of software development budgets) are reduced significantly, often as much as 75%.*

*Visible helps its clients become world-class software developers. With our assistance, clients improve processes and procedures until they are able to consistently develop quality information systems that exactly meet their own and their customers' critical information needs.*