

USoft Product Information  
USoft Definer

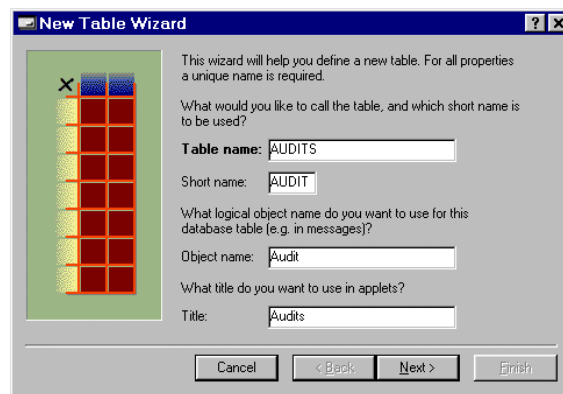


## USoft Definer Creating Rules-Based Applications

### Business Rules

The power of the USoft environment is its ability to create and customize adaptable application versions from application models. These models may be relational or based on the Universal Modelling Language (UML). USoft supports the Dynamic Systems Development Method (DSDM) and other popular Rapid Application Development (RAD) standards.

The cornerstone of USoft technology is that the application is built from business rules. USoft Definer allows developers to create and maintain application models and business rules quickly and easily.

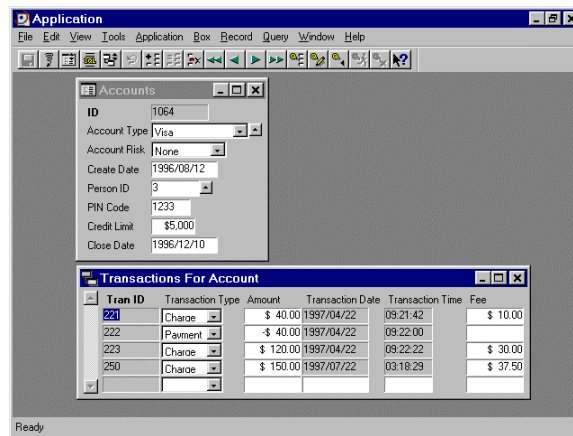


Models can be defined in-place, or they can be imported from an external modelling tool using the USoft Data Model Loader.

USoft Definer has wizards that make application definitions easy. Once an application is described in the USoft repository, behaviour rules can be added on. These rules are instantly translated into runtime behaviour in prototype applications.

The Definer will create the native database schema based on the model definition, and it will continue to manage changes to the schema as the model is being changed in USoft. Prototype applications are automatically available, allowing users to verify functionality before development time is spent making customized user interfaces:

- For every table, a fully functional user window is created.
- For every relationship, a related window is created for each child table. For example, if a model where an Account has many Transactions, then a related, multi-record Transaction window will be created that is automatically linked to the Account window.
- For every foreign key, a lookup button is created which gives access to a lookup window displaying parent data.



### Prototyping

End users can evaluate recently created functionality using the prototype applications available during each development phase. This allows them to identify design mismatches at an early stage. Behaviour declaratively specified in Usoft Definer - uniqueness checks, referential integrity, mandatory values, etc. - is automatically enforced in the prototypes and can be rolled back at any later stage. Usoft refers to this behaviour as implicit business rules.

Prototypes can be gradually enhanced by adding explicit business rules. These rules are written in standard SQL or in a third-party component, and are automatically tied to the core application model. They are automatically enforced in the runtime application – you do not need program behaviour or explicitly tie it to interface events. In all its tools, Usoft provides a graphical SQL editor to help you specify rules in SQL.

### Rules-Driven Method Invocation (RDMI)

In addition to implementing application-specific business rules in SQL, you can have the runtime Rules Engine call external COM+ and J2EE components and Tuxedo services and Web services. This provides open standards for runtime architecture while at the same time allowing central maintainability of your business rules.

### Features

- Definition of application models and explicit business rules.
- Repository-based development. The repository centrally stores and re-uses data model definitions, business rules, user interface definitions, security and user information.
- Business Rules-based development. Application behavior is formulated and maintained in the form of maximally modular and adaptable business rules.
- Fully functional prototype applications for model and business rules verification by end users at every stage of development.
- Meta-data import from any Microsoft Repository or Universal Data Access (UDA) source – this includes virtually all industry-standard modeling tools and data dictionaries.
- Usoft's Web Services solution allows other applications to communicate with Usoft applications using Web Service technology.
- Supports iterative Rapid Application Development (RAD) methods including the Dynamic Systems Development Method (DSDM).
- Supports import, export, and querying of XML documents.
- Multi-language specification of messages and user text.
- Printing capabilities in the production (runtime) environment.
- Wizards to help you define data model elements and business rules.
- Graphical SQL editor to help you define rules using the SQL language.

### **Benefits**

- Single point of definition for all application behavior, decreasing the development and maintenance cost for database applications.
- Rapid, short and manageable development cycles, allowing constant feedback to management and future end users.
- Rules-based definitions allowing maximum modularity and adaptability.
- Interoperability with external modeling tools, legacy application definitions, and runtime third-party components.
- Increased productivity through high development-time usability: Easy-to-use interfaces with built-in wizards to perform basic or recurrent tasks.
- Open architecture, via COM+, J2EE, Web Services, and import and export of XML.