SLINGSHOT™

eTools - Rapid Application Development Workbench

All Slingshot Software applications are developed using eTools. eTools is the key that unlocks control of your enterprise business systems. The eTools workbench allows you to quickly deploy ERP software and reduce ongoing maintenance costs.

Use eTools to quickly adapt Slingshot applications to your specific requirements. Don't compromise; optimize your applications to improve your competitive position. What's more, your changes are clearly identified. The eTools "Upgrade Assistant" allows you to merge these changes into future releases of the Slingshot's applications.

Slingshot's eTools module provides many unique advantages, as outlined below:

eTools empowers you to customize and extend your own applications without programming. eTools lets you take control of your enterprise business systems, increasing flexibility and scalability while decreasing cost and risk.

Provides a "Tailored Solution" at a Package Cost

eTools empowers you to extend your Slingshot applications to meet your unique requirements. Most importantly, it allows you to do this without software vendor support. Consulting fees are minimized; application fit maximized.

Shortens Application Development Time

*e*Tools allows industrial strength applications to be developed in hours rather than days, weeks or months. The developer simply updates the repository and *e*Tools automatically generates and compiles the required code.

Reduces Ongoing Costs

With eTools you develop self-sufficiency never before possible with a tailored solution. If you need a new form, report, menu, role or data table you can create it – quickly. The Upgrade Assistant utilities also allow you to take advantage of future versions of Slingshot's products, without having to manually redo all of your personalizations.

Ensures High Quality Code

*e*Tools templates generate bug free code. It's that simple, because all objects have been established and validated within the repository.

The eTools Approach

The secret to eTools is found in how it uses Object-Oriented Programming concepts (inheritance and encapsulation), to rapidly generate applications. Data fields and formulas are grouped into classes, and forms are defined for each class. Menus can be set up that present a hierarchy of forms to each user based on the user's current "Role". This information is all maintained in the eTools Repository.

eTools uses code generators to create the application. Form definitions result in programs that generate HTML; code is subsequently generated to transfer data from and to the database. Finally, commands in the editor create databases, tables, columns and import and export data. All generated code utilizes Microsoft's latest architecture ".NET" making eTools a non-proprietary solution.

Special business rules (methods) determining your system's functional logic can be developed and executed at the appropriate time. These business rules may be developed using Visual Basic, C++ or any other programming or scripting language. Each rule is registered in the repository along with a description of its interface (arguments). This simplifies the process of reapplying your rules to future releases.

"Built In" Application Services

In addition to generating the application, eTools provides a number of "services" important to most applications.

- **Security Infrastructure:** A set of forms is provided for the Security Administrator. New users can be added and assigned one or more User Roles (described below). Stringent password security and system access management is enforced.

- **User Roles:** Form and menu definitions can be varied by user role. A user can be authorized (by the Security Administrator) to perform one or more roles. The user can switch between roles within the application.

- **Span of Control:** Users can be restricted to certain information based on Span Of Control definitions set by the Security Administrator on the user record. For example, you may decide to restrict access to inventory information based on the User's warehouse. Or, restrict access to financial information based on Entity and Division. .eTools allows any field to be used for span of control purposes. Simply set the field's "Span" attribute and define the forms to which the restriction applies.

- **Regional Preferences:** Regional preferences include language, date mapping and monetary (multi currency) masking rules. Each user can request a regional preference. Labels, lists, dates and monetary amounts are automatically shown in the users preferred language and format.

- **Alerts:** Alerts monitor exception conditions, prompting notification (and on occasion automated remediation) as needed. For example, you may want to monitor perishable inventory that is about to expire. Alerts are defined in eTools and include a query to test for

the condition, a message to display to the user when the condition is true, and an optional form to display details.

- **Label Translation:** A dictionary in eTools is used to translate labels, messages, menu items, form titles and static strings into multiple languages. This information is used to generate forms in each supported language.

- **Reports Infrastructure:** Reports can be defined in Crystal reports and integrated into any application. Report initiation forms can be developed in eTools. These forms accept parameters and launch the report writer. Queries defined in eTools, can be used for record selection and sorting. An ADO schema presents your applications "object model" to the Crystal Report. Formulas defined in the repository (computed fields) can be printed on any document or report. All reports can be viewed online, printed in background, uploaded to Excel or Word or, exported in Adobe portable document format (pdf). Network printers can also be registered with a network path and native drivers.

- **Control of Inactive Records:** A reserved field name (IsInactive) is used to flag inactive records. If the record is inactive it remains in the database but is suppressed (by default) from all search forms and combo boxes. The application designer can allow certain "Roles" to reactivate inactive records.

- **Data Audit Trails:** Any field can be flagged as an audited field. If so, all changes are written to a separate audit log. The log includes the transaction (created, changed, deleted) the "before" and "after" values, the date and time the change was made, and the user. A default form is created automatically allowing you to search for changes by field name, user, date etc.

- **Data Archival:** Information that is no longer active can be copied to a separate archival table. Simply flag the class as archived and define the condition. A reserved field name (IsReadyForArchive) contains an expression indicating whether or not the record should be moved to its archive. eTools generates the necessary tables and the program to move information to the archive for you.

- **Process Monitor:** A process monitor facility a background task queue and executes jobs as needed. Each job can be launched immediately, scheduled to execute at a certain time, scheduled to execute each day at a specified time or, scheduled to execute each week on a certain day of the week at a certain time. The job queue is available to authorized personnel (defined by User Role). Jobs can be removed from the queue or rescheduled.

Upgrade Assistant – Simply Preserve Your Personalizations

The update wizard compares a new version of the new application to your current version and allows you to reapply the changes you've made automatically. These changes can include changes to forms, menus and tables – even business logic. This allows you to extend your applications and reapply your changes when the next release is issued. Only Slingshot allows you personalize your applications and retain vendor support.

Object Oriented Approach Simplifies Repository Maintenance

The eTools repository allows classes to inherit characteristics from other classes or data types. For example, "requested shipment date" is derived from the "date" type and thereby automatically inherits that type's traits.   The rules for editing, formatting and access can all be inherited. This allows you to change properties at one level and have the change reflected in all derived classes.

A Non-Proprietary Approach

*e*Tools generates Visual Basic .NET code, and an ActiveX Data Objects (ADO) compliant database (SQL Server or Oracle). This non-proprietary approach provides the assurance of Microsoft support for all technical aspects of the application.

SLINGSHOT™

Over the past 3 decades, Slingshot has helped companies just like yours achieve the next level of efficiency, growth and profitability.

Our innovative approach to ERP software development, implementation and ownership allows you to extend your ERP systems without programming.

This lowers your costs, ensures scalability, decreases time to return on your investment and lowers your overall risk.

When it comes to ERP software – we innovate.

Visit us online:
slingshotsoftware.com

Slingshot Headquarters
270 Bridge Street
Suite 302
Dedham, MA 02026
781.329.1900 – Phone
781.461.2421 - Fax

Slingshot – Western Region
Xerox Centre
1851 East First Street
Suite 900
Santa Ana, CA 92704
714.427.1170 – Phone
714.427.1177 – Fax

Sales
Toll-Free: 1.866.754.6474
sales@slingshotsoftware.com

Customer Service
1.781.329.1900 (X206)
service@slingshotsoftware.com

Marketing
1.781.329.1900 (X205)
marketing@slingshotcity.com