

# OpenROAD 6.2

Simpler Restyling, More Customization, Faster Deployment and Less Code

## Key Benefits

Codeless 1-pass application restyling

Unlimited extensions and customizations

Reduced custom runtime code and startup time

Faster application customization

More functional and faster display generation

Simplified migration, test and deployment

Developers using OpenROAD 6.2 will find a much improved platform that can easily restyle existing applications, extend and customize business objects, field types, and frame types, and add new field and frame behaviors (like pulsing) that either required a lot of code or simply weren't feasible before. Coding is simpler and more comprehensive. New user class and display generators allow developers to leverage the tremendous amount of information stored in a typical enterprise database when developing end-user facilities involving interconnected business objects. OpenROAD 6.2 also includes Loadrun, a big step forward in making it simple to deploy OpenROAD applications.

## Codeless 1-Pass Application Restyling

OpenROAD 6.2 developers now have the tools to restyle an existing application to a targeted look and feel without writing any new code. Property Changer lets the developer modify the entire application, parts of an application, or individual frame transformations. A Windows 7 restyling template has been added to the Core Library. The Property Changer wizard can step you through a simple process to pick the application or applications and the style transformation and run it against the source (after your existing application has been backed up). Property Changer is a non-OS specific, generic framework that can be used to restyle the application not only with OS features but with, for example, branding features, such as changing a product name.

## Unlimited Extensions and Customizations – Tagged Values

Tagged Values, new in OpenROAD 6.2, allow developers to customize OpenROAD much more richly, and with simpler code. Tagged Values are name-value objects available in every field, component, application, attribute and method. Existing business objects, field types, frame types and database interactions can be extended indefinitely without worrying about polluting the core definitions. Tagged Values let developers reach the areas that database definition and object-oriented structuring can't reach. With the new sprites and input/event/response processing, field and frame appearance can be much richer and match chosen styles.

## SetupFrame

The new SetupFrame saves custom runtime coding and reduces startup time. SetupFrame lets developers predefine frame setup code and put all setup code in single, coherent locations separate from the runtime application. All the changes to the frame are applied in advance so that when a frame starts up at runtime, everything is ready to go – significantly reducing startup time and reducing the complexity in managing frame setup.

## Customizable Field and Frame Behaviors

Prior to OpenROAD 6.2, the many style and guidance features required or expected by end users either took a lot of custom code or weren't feasible. The new field and frame 'defined behaviors' bypass the need to use event-driven 4GL code for these features, instead reacting directly to end-user actions and timed alerts with a rich and customizable set of responses, including visual property changes and animation capabilities. This approach, of which the new Windows 7-like styling is an example, dramatically speeds up application customization and keeps the event-driven business logic free of display code.

## User Class and Display Generator

OpenROAD 6.2 exploits the business information stored in a mature database with new user class and display generators. The new generators take a heuristic approach to database indexes, data distribution, relationships and constraints as well as conventional table and column definitions, and the generated user classes include much ancillary information, stored in TaggedValue sets and QueryObject definitions. User class generation can be cascaded to include all classes related even indirectly to the seed user class, and the result is a much richer, more accurate and crossreferenced set of user classes, where every class is not only attribute-complete in itself, but also connects to related business classes. The variety of displays that can be generated from and interact with these user classes will make business sense and have substantial functionality out of the box because the user classes are better defined. The generators are richly customizable, and the generated user classes and frames are readily and robustly editable.

## Deployment - Loadrun

The Loadrun tool simplifies and manages deployment of all OpenROAD applications. Loadrun allows deployment from a central location to remote machines – eliminating the need to physically install it on each individual machine. Once the OpenROAD runtime is installed on a client, a centralized Loadrun server can manage deployments and updates to any client machine. This includes side-by-side, segregated deployments of different versions of OpenROAD and/or applications – making migration and test much simpler.