

Frame Gallery - General Information

The following topics are covered below:

- What is the Frame Gallery?
 - Benefits Provided by Frame Gallery
 - Application Development Procedure
-

What is the Frame Gallery?

The frame gallery and the application shell are development tools for dialog applications. The main components of the frame gallery and the application shell are:

- **Frame gallery**
Using the frame gallery, you can select one of a range of standard dialog types (or frames) and generate a simple dialog and associated data access modules. You can then use the dialog editor to customize the generated dialog to include application-specific validation and other processing needed for a fully functional application. Suggested code in the generated dialog assists you with the customization process.
You can use frame gallery to generate prototype dialogs and functions during the product design phase to see what an application will eventually look like. Frame gallery is primarily used, however, to generate dialogs and functions during the implementation phase.
- **Application frames**
The frames used in the frame gallery include large amounts of standard internal logic to handle browsing, command processing, navigation between dialogs and database access. In addition to these frames, further frame code is available which you can include in your application to handle various special requirements.
- **Application shell**
The application shell provides an application framework which you can use to run dialogs and applications developed using the frame gallery. It also provides a range of administrative functions used in application development (for example, application definition) as well as in maintenance (for example, user maintenance).

Benefits Provided by Frame Gallery

The frame gallery contains all components of a dialog system that are not application-specific. Into this generic system you add your application-specific functionality.

Using frame gallery provides the following advantages over conventional application development:

- **Reduced implementation requirements.**
Standard functions need not be individually coded for each new application. The developer can concentrate on the often very complex application-specific requirements.
- **Reduced testing requirements.**
The frames provided have already been tested and are error free.
- **Easy customization to meet application-specific requirements.**
In addition to basic functionality, most frames also contain suggested code which can be used to customize dialog functions to meet application-specific requirements.
- **Protection of investment in application-specific code.**
Standardized logic and application-specific code are carefully separated in dialogs through the extensive use of copy code for the standardized frame logic. This means that upgrades to frame logic in future versions of the Frame Gallery can be easily incorporated by simply restoring existing dialogs.
- **Easy system orientation and maintenance for developers.** Because a similar structure is used for all frames, it is easier for all developers to become acquainted with any given functional aspect of the application. This reduces orientation and maintenance requirements significantly.

- Standard application navigation for end users. End users are always provided with the same applications structure, which increases system acceptance by the user while at the same time reduces user orientation and training requirements.

Application Development Procedure

The purpose of this section is to describe how an application could be created using application shell and frame gallery. The information provided is meant to be a guideline and not a complete recipe for application development.

The procedure below is a general recommendation for creating applications.

1. Write application specification and analyze requirements.
2. Define user interface standards and naming conventions.
3. Define and/or generate database schema and database definition. Decide which entities are to be implemented as tables.
4. Familiarize yourself with application shell and frame gallery.
5. Structure the application by determining what types of dialogs and functions are necessary.
6. Assign a frame to each function.
7. Define a start application in the application shell and specify a Natural library for the application.
8. Generate prototype dialogs and functions.
9. If not already performed in step 3, define and/or generate database schema and database definition. Decide which entities are to be implemented as tables.
10. Define tables.
11. Generate production dialogs and functions.
12. Customize the application.
13. Set up icon-based navigation.