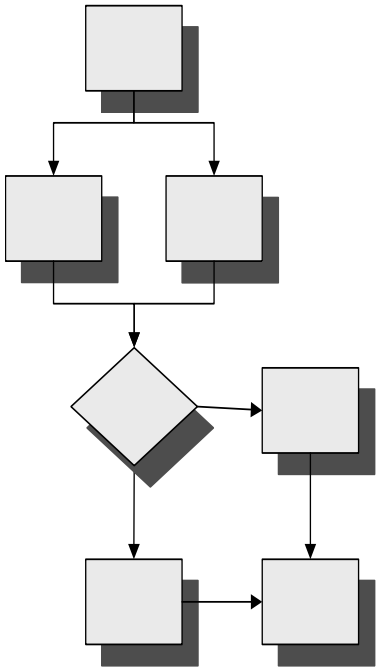
Control Business
SOLUTIONS IN TECHNOLOGY

WHITE PAPER

For Developers



DEVELOPMENT OF APPLICATIONS AND INTERFACES FOR INTEGRATORWP®

IntegratorWP® V1.0

1. Creating Applications and Interfaces for IntegratorWP®

IntegratorWP® is a workflow based platform for the development of integration and system applications. The concept of a *workflow platform* for development is beneficial to the majority of applications based on centralized and automated workflow processes. Applications gain agility when a platform with pre-existing rules is utilized, thus increasing the capability to integrate with other systems that use Web Services technology, an industry tendency, or other existing integration standards through the development of specific interfaces.

2. IntegratorWP® Platform

The **IntegratorWP®** platform is composed of three layers, as shown in figure 1.

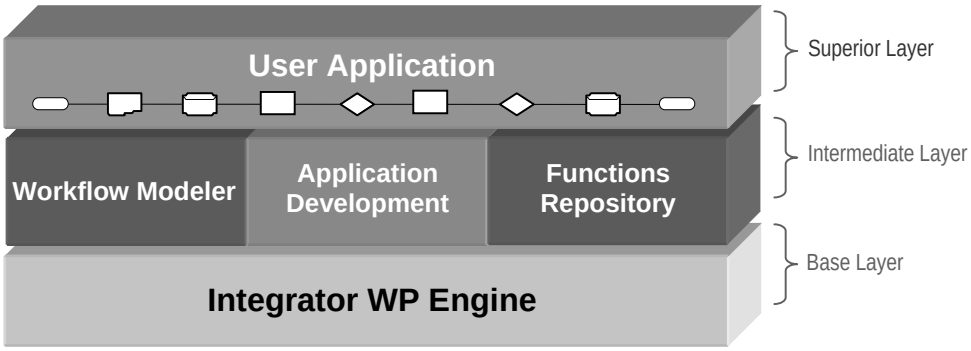


Figure 1

The superior layer processes customized applications. The intermediate layer, which houses the application development tools, is sub-divided into the following modules: Workflow Modeler, Applications Development and Functions Repository.

3. Functions Repository

Functions are programs that when combined form an **application**.

The **IntegratorWP®** architecture possesses a single library that stores all of the functions, developed by the client or existing system functions that are already available for the developer's use.

See the examples below in Figure 2.

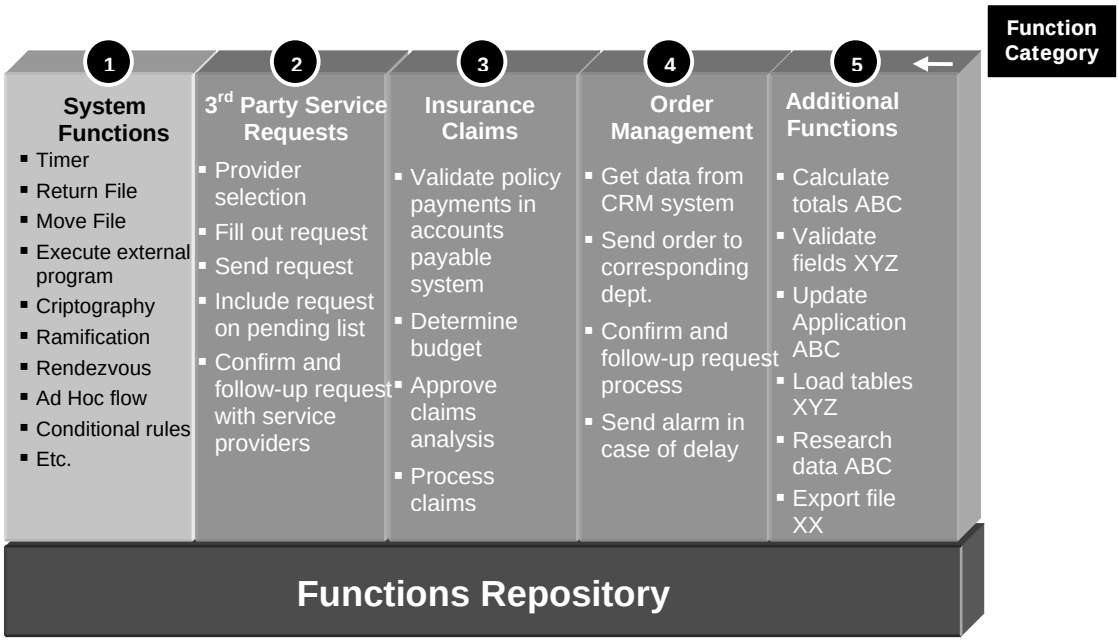


Figure 2

The functions are grouped in categories as follows:

Categories of System Functions: group functions written in C# language and, VB.net. These functions are available to developers to customize via *function parameters*, which are an integral part of each function.

Categories of User Functions: are groups of functions that have been coded by the client or development companies. These categories group functions that combined may form an application or may be used independently. They can be written in VB Script or Java Script codes.

4. Workflow Modeler

Figure 3

The **IntegratorWP®** workflow modeler has two primary objectives: the design of the workflow and the creation of forms. The design specifies the route and the rules that a workflow will follow from the beginning of the process to its conclusion. Forms are utilized for data entry, as well as, updates of existing data in the system.

Workflow Design: To create the workflow design, **IntegratorWP®** provides an environment that is 100% graphical. The *workflow engine* utilizes Microsoft's MS Visio®, one of the strongest graphical design tools that is currently available.

Creation of Forms: The creation of forms is also developed in an environment that is 100% graphical. In this case, **IntegratorWP®** utilizes Adobe Acrobat® 6.0 as the engine that builds the forms, from the initial construction to the daily maintenance activities that may be required.

5. Developing and Compiling Functions

Compiled Scripts: The **IntegratorWP®** development environment permits the creation of functions in VB Script and Java Script on .net, thus allowing the written codes to be **compiled.** This creates executable programs that on average present performance improvements up to 40 times faster than interpreted scripts. The compiled script capability enables the construction of large and complex applications within **IntegratorWP®** that support large amounts of data without having a negative impact on performance when in the production environment.

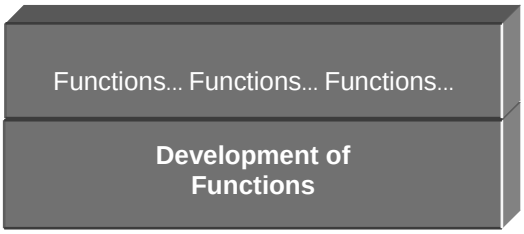


Figure 4

VBA Toolkit: IntegratorWP® provides the development environment VBA Toolkit that utilizes VB Script.net. Since VBA Toolkit is a very user-friendly development tool, it quickens the function's creation, in addition to, facilitating eventual future alterations if necessary. Pre-existing customer Visual Basic programs may be utilized with the product.

Integration: IntegratorWP® permits the integration between external applications, third party products and/or legacy systems for consultation or functionality interaction.

6.Function Parameters

Parameters: Within **IntegratorWP®** functions may be reutilized through the concept of *function parameters*, and variables may be introduced according to specific requirements.

The objective behind *function parameters* is to avoid cases where a functional necessity serves two or more applications, and that the function is replicated and customized to satisfy the variable requirements of each case.

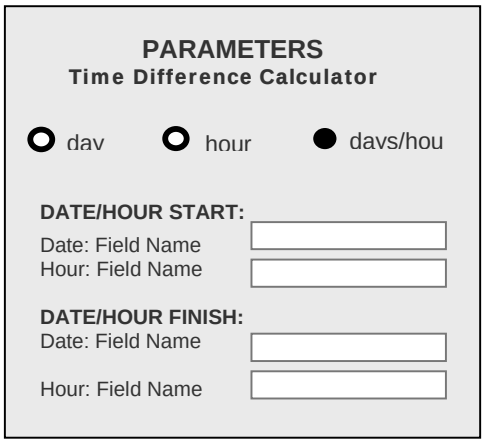
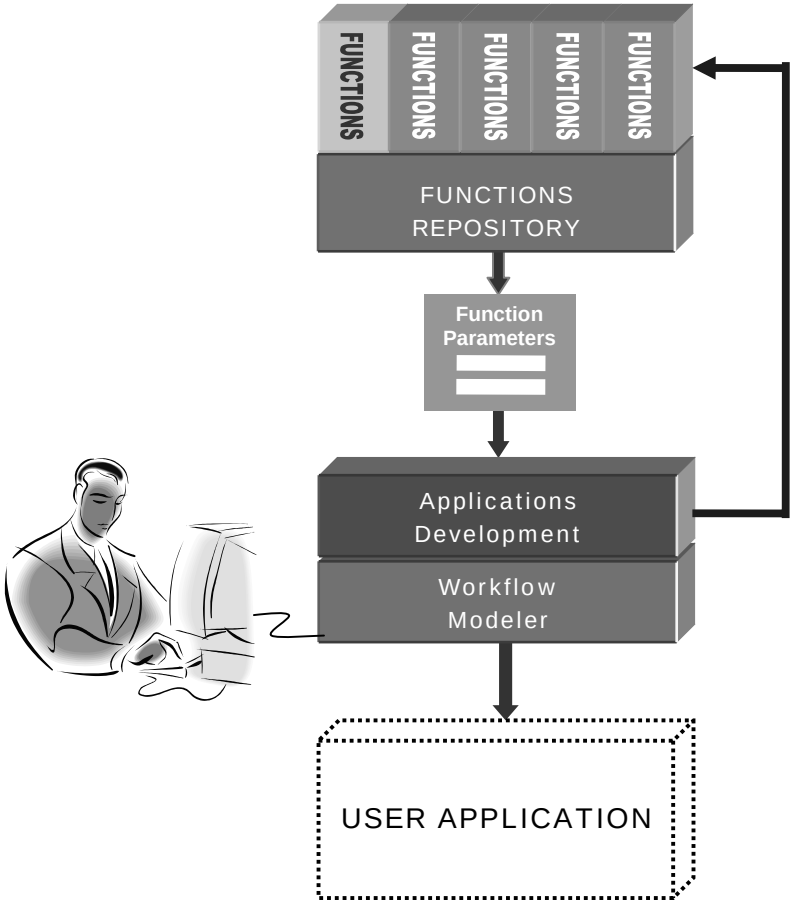


Figure 5

In the example Figure 5, the function can determine the difference in dates by calculating time in days, hours or days/hours without having to alter the original function code. The *function parameter* feature is available for use by both, system functions, as well as, user functions.

Application Construction

In Summary, the **IntegratorWP®** components that are used in the construction of customized applications are: the workflow graphical modeling and forms creation environment, the compiled functions development environment and the functions repository.



Conclusion: Figure 6 presents a diagram that demonstrates the steps in creating an application. The developer models the workflow, creates the forms, utilizes the functions available in the function repository, defines the parameters and builds additional functions with the applications development module.

Control Business Technologia, Ltda.
Avenida Jamaris, 100 - Conj. 802
São Paulo, SP Cep 04078-000 Brazil

Telefone: +55-11-5053-3565
www.controlbusiness.com.br