

Benefits:

- Automates access to closed applications
- Reduces development time
- Simplifies ongoing maintenance
- Enables practical development of complex integrations

Features:

- 100% data-driven and configurable
- Local, single session version or remote, multiple session version.
- Enables integration modifications without code changes
- Expandable with custom code if needed

Most Applicable:

- Rapid development is required
- Application has no or limited API
- Process includes many paths / exceptions
- Target applications change often

Turn Any Microsoft Windows® Application Into A Web Service

Introduction

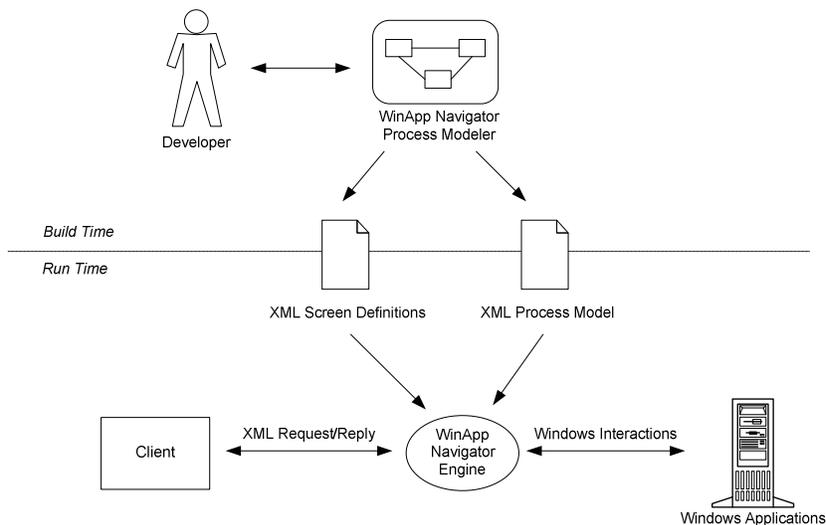
Almost every corporation deploys Microsoft Windows® applications in their enterprise. As stand alone entities, these applications are largely very successful. A major problem emerges when these applications lack an application programming interface (API) which greatly reduces the opportunity for process automation and enterprise application integration (EAI). To address this challenge, Wazee Group has developed the WinApp Navigator™ which allows an XML-based interface to be easily developed for any application. With this tool, any Microsoft Windows® application can be accessed via an Xml interface and, with little time and effort, turned into a web service.

Overview

WinApp Navigator™ addresses the issues with applications that have no APIs. This tool provides the ability to logically name relevant items in the graphical user interface (widows, controls, etc) and use them to create graphical process models of the integration process.

The visual representation makes the complex navigation paths far easier to understand, and when screen or process changes occur, developers need only to modify the graphical process model. This provides an environment that simplifies integration development and maintenance, because all navigation paths can be seen and changes can be made rapidly, often without code modifications.

During the build phase, the tool converts the graphical process model into an XML file and a file containing the screen definitions. During run time, a XML request is sent to the run time engine that accesses the process model for the request and the screen definitions. The engine then executes the model via terminal emulation to the legacy system. At the completion of the process model, a XML reply with a status and the resulting data is returned to the client.



(continue)

Graphical editing of process models.

Create transactions to the applications without writing a line code.

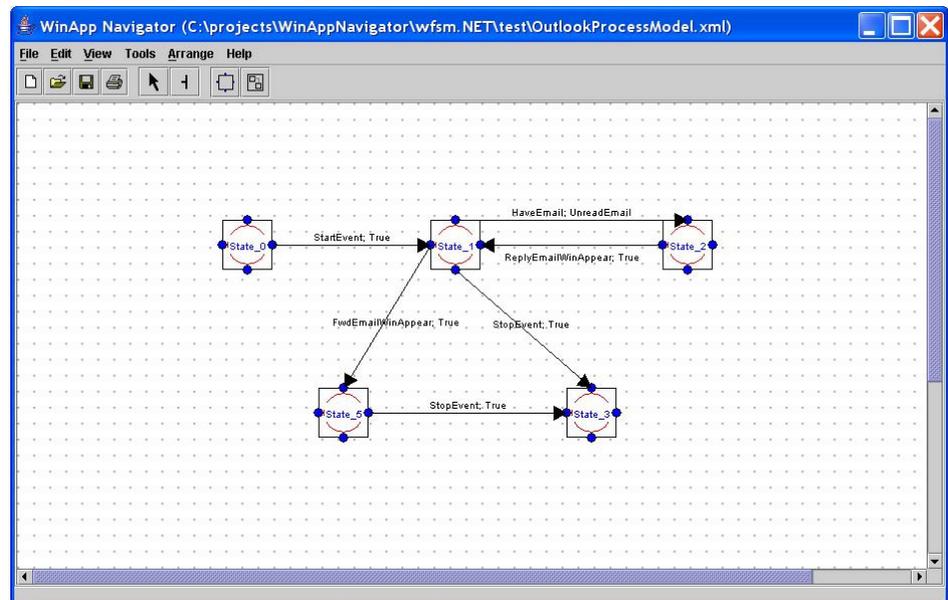
XML based data allow modification via the tool or manually editing the files.

Modeling Process

Using WinApp Navigator™ involves a six-step development process:

1. **Define Windows:** identify windows and controls.
2. **Describe Events:** define what items or changes within a window will drive the integration process using pre-built components or extend functionality through custom built .Net classes.
3. **Develop Actions:** establish what will be executed in response to an event using pre-built components or extend functionality through custom built .Net classes.
4. **Create Conditions:** describe Boolean conditions to be evaluated before processing actions.
5. **Model Process:** graphically depict integration using above components.
6. **Execute Process:** start and manage execution utilizing the tool's API.

This information is captured in two Xml files, one for the windows and controls definitions and one of the process model. A simple model is shown below:



Summary

WinApp Navigator™ provides automation and integration possibilities where there are none. With the graphical process editor, the tool transforms these possibilities into realities in little time and with little effort. WinApp Navigator™ provides the critical building blocks for turning any application into a web service which can be leverage throughout the enterprise.