

***Insurance on Line***

***IOL***

***Technical Overview***

# Contents

<b>INTRODUCTION</b> .....	<b>3</b>
<b>INSURANCE ON LINE (IOL)</b> .....	<b>3</b>
OVERVIEW .....	3
IOL FRONT-END GUI SCREENS.....	3
IOL DOT NET SERVICES .....	4
IOL DATABASE.....	4
IOL REPORTING.....	4
IOL WORKFLOW AND ALERTS .....	5
IOL EXTERNAL INTERFACES .....	5
IOL NEW INSURANCE PRODUCTS.....	5
IOL SOURCE CODE .....	6

## **Introduction**

The document is intended to provide a technical overview of insurance on line and the underlying technologies. It is presumed the reader has an IT background and understands core Microsoft technologies such as Windows, Dot Net, Internet information server and Microsoft SQL server.

## **Insurance On Line (IOL)**

### **Overview**

IOL has been developed from the ground up based on the latest Microsoft technologies. It is a 3 tiered client-server application based around:

- Microsoft Dot Net web based front end
- Microsoft Dot Net services
- Microsoft SQL database server

IOL is built on the KIS principle (Keep It Simple). By isolating and utilising each of the 3 tiers of the client-server model complex coding has been avoided in the core application without having to compromise features.

This combination of technologies provides for

- Web enabled
- Security
- Rapid deployment
- Scalability
- Performance
- Customisation
- Open-architecture reporting
- Strong integration with the Microsoft platform

### **IOL Front-End GUI Screens**

IOL screens have all been developed with Microsoft Dot Net as web pages. The front end screens primarily capture data. All validations and data-I/O requests are passed to the appropriate Dot Net web services that in turn call the SQL database. Advantages of this model are:

- Ease of deployment – each client requires only a web browser
- Ease of administration – The application is deployed once to a central server(s)
- Ease of development – Front end merely for data-capture, no complex logic
- Ease of customisation – New fields can be added to front-end and simply linked to back end SQL database
- Lower cost of hardware – Minimal requirements at client to run internet browser.
- Scalable – For increased performance and throughput simply upgrade web server or SQL server as appropriate.

If required, programs can be re-compiled as Visual Basic applications.

## **IOL dot net services**

The IOL front-end GUI screens call Microsoft Dot Net web services.

Dot net services provide the following features:

- Basic validations for the front-end GUI
- Data requests formatted passed between the front end and back end database
- Other applications can easily call the dot Net services

## **IOL Database**

IOL uses Microsoft SQL server for its underlying database. Because IOL has been written from the ground up it makes full use of SQL server. This includes:

- Stored procedures
- Triggers
- Functions
- User defined validations
- User defined business rules
- Data validation at the table level
- Data auditing at the table level
- Scheduled tasks
- Email alerts

The IOL web services call IOL stored procedures on the database to perform I/O functions and enforce business rules. Stored procedures return appropriate datasets back to the web services or an error message.

For 'update procedures' the stored procedure enforces business rules, performs validations and calls user written stored procedures for customised business rules. If the data passes these validations a table insert/update/delete is attempted.

SQL triggers then perform table level validations and make audit copies of old records as appropriate.

## **IOL Reporting**

IOL uses a combination of crystal reports and Microsoft Excel for reporting. Reports call SQL stored procedures that return datasets. This maximises performance of the database and LAN/WAN. Crystal reports can be run in the dot net framework and delivered to users as web pages.

Subject to appropriate security users are encouraged to develop their own reports to suit their specific business needs. These can be developed with Crystal reports, Excel or their preferred tools.

## **IOL Workflow and Alerts**

IOL uses in-built SQL features to manage workflow with the ability to create set up tasks such as automated emails/faxes to advisors, customers, insurers and underwriters.

Tasks such as specific business alerts and data integrity checks can be defined and implemented. Periodically emails can be sent to appropriate users for events such as 'New business signed for over \$X', 'Policies in suspense that require activation', or 'policies awaiting payment'.

Some system tasks are automatically performed by the SQL job scheduler such as in-force year generation and synchronisation with the accounting system.

## **IOL External Interfaces**

The system is based around open architecture and therefore can be accessed by other applications. These may be for reporting or integration with other applications such as CRM and financial systems.

IOL has a single transaction file that contains financial information and provides a single point of integration with financial systems. This interface can be either for the General Ledger or Accounts receivable.

IOL currently interfaces with Microsoft Great Plains Receivables (AR) for policy invoices, debt collection and cash receipting, and Payables (AP) system for advisor commissions and claims.

## **IOL New Insurance Products**

IOL has been designed to allow rapid deployment of new insurance products. The base system contains information appropriate to all policy holders and all insurance products such as premium, name, address etc.

Specific data entry screen are developed for each product. For example, personal accident requires specific information such as job class, weekly income etc whereas Motor vehicle requires car make, model etc.

The following steps are required to add a new insurance product to the system:

- Add new product to the product file
- Define date required to be captured for insurance product
- Develop GUI screen for front end data entry
- Create SQL table
- Define and implement business rules for insurance product
- Design and develop report for renewal notices

## ***IOL Source Code***

Subject to appropriate agreements IOL comes with the following source code for modification during implementation by Orion or the client:

- Dot Net GUI front-end screens
- SQL objects such as stored procedures, triggers and business rules.
- Crystal reports
- Microsoft Excel reports