Transaction
Re-engineering
Using
Scanning
Technology

Dissertation submitted to the Institute of Technology Sligo in part fulfilment of the requirements for the award of Masters (MSc) in Computing

G. M. Farrelly

2005

Table of Contents

Table of Contents (cont'd)

Declaration

I hereby certify that the material submitted in this dissertation, towards the award of **Masters (MSc) in Computing**, is my own work and has not been submitted for any academic assessment other than part-fulfilment of the award named above.

Signature of candidate:............Gerard..

Date:..........15/7/05..........................

Acknowledgements

Very sincere and heartfelt thanks to the Directors, Management and Staff of Excel Industries especially Tom and Barry.

Thanks to my colleagues in the Department of Electrical Services Engineering in particular Colm Agnew, Brian McNally and Tommy Shannon who, whenever it was asked for, gave advice and assistance willingly.

A special thanks to my supervisor Dr Paul O'Leary whose guidance, common sense and good humour ensured the completion of this task.

Finally, thanks to my wife Linda and my son Martin for keeping me happy and letting me get on with the work.

Foreword

The title of this project derives from a play and a movie that share the same name. The play (Proof - David Auburn) and the movie (Proof - Jocelyn Moorhouse) share a central thesis that proof comes before trust in personal relationships, and it would seem that this is true of business.

A pivotal driver in the success of any business is cash flow and a principal component of this is the tedious and tiresome function of credit control. Despite the idealistic concept of business based on trust most business is in fact based on proof; proof of delivery, proof of receipt, proof of payment.

Chapter 1: Overview

1 Overview

1.1 Introduction

The project arose out of a need to update and improve the overall efficiency of a small/medium sized manufacturing and distribution company.

The re-engineered system streamlines the process of ordering, production delivery and accounting. In addition, it increases end-user involvement in the networked processing system. The project was separated into two phases.

- *Phase 1:* Re-engineer the business processes as part of a cost benefit analysis.
- *Phase 2:* The development of a document imaging system, which could be integrated into the networked transaction processing and accounting system.

This involved the utilization of bar coding as a unique identifier for each order/delivery docket. Documents are scanned and the data stripped out and stored as simple data files in a database. This speeds up the process of data storage and retrieval and facilitates access to the data by other functions within the organization.

1.2 My role

My role in the project was that of project manager charged with overseeing the project from initiation to completion. I reported directly to the Financial Controller and Board of Directors of Excel Industries. As project manager I was required to carry out the business systems analysis in conjunction with Excel Industries staff, liaise with consultants, develop the programming code and supervise the implementation of the project.

To these ends I was required to undertake the following tasks:

- Seek out potential systems providers
- Develop the specification
- Test and validate the new system
- Tailor the new system to conform to the BASDA specification
- Develop and test the XML code for systems integration
- Ensure training of staff
- Plan and manage the upgrade of the computer network

At various times in the life of the project I was called upon to make recommendations and decisions of a "go/no-go" nature. These recommendations included, selecting the providers of the document managing system, opting for the BASDA schema, and choosing the methodology for the systems development and deciding when it had reached a sufficient level of maturity. The final decisions at all stages were reached in conjunction with the Excel Industries management and board.

1.3 Synopsis

Chapter 2

This chapter outlines how the project arose, what it hoped to achieve and the approaches adopted to fulfil these hopes. The chapter contains a description of the principal features of the new system and the structure of the manual systems employed in handling customer queries heretofore.

Chapter 3

A brief description of the typical development models that are extant and the reasons why an evolutionary approach was chosen for the project are outlined in this chapter.

Chapter 4

This chapter recounts a detailed description of the prototyping that took place in the course of the requirements analysis for the project. It also shows how this activity and some of the unforeseen difficulties impacted on the final working version of the system. The decision to opt for the evolutionary paradigm outlined in Chapter 3 was more than justified by the fact that the final version differs significantly from the original brief. For example, it was never envisaged at the outset that the new Document Management System and the Exchequer Accounting application that Excel Industries used would be integrated.

Chapter 5

Chapter 5 relates in detail the components of the developed system. These include inter alia; bar code generation, the underlying XML schemas and the stylesheeting code required for transforming the presentation.

Chapter 6

An overview and schematic representation of the new system are presented in this chapter. The ADOS platform and its associated functionality are described and the sequence for scanning and uploading documents via the user interface is also detailed. The latter part of the chapter relates the network infrastructure upgrade that took place as a consequence of the decision to implement the document management system.

Chapter 7

This chapter contains a critical review of the project in the context of the original objectives and the final outcome and finally goes on to consider the possibilities for the future.

Appendices

The information contained in the appendices includes details on bar code verification, the BASDA schema and the XML/XSL code developed for the project.

Chapter 2: Background and Context

2 Background and Context

2.1 Introduction

The genesis of this project is rooted in the difficulties that manifested themselves whilst Excel Industries were operating its paper-based system. This system led to the generation of inconsistent and redundant information. The need to allocate staff time and effort to the organization of the information and the allocation of valuable floor space to the storage of documentation constituted a significant overhead.

Queries directed to more than one of the business functions might result in contradictory responses. For example, a customer query alleges that the full complement of requested items is incomplete and contacts the company twice. On the first occasion s/he is directed to Distribution and Distribution dispatches the alleged missing items to make up the shortfall. In the meantime the customer follows up the initial query and on this occasion accesses the Accounts section and the response of the staff in accounts may be to extend credit based on the non-receipt of goods.

In attempt to address this issue the delivery dockets were redesigned as multi-page (4 part) format. This enabled copies to be stored in each of the main operations within the company. This however, resulted in greater demands for storage space. Each division could now handle queries directly based on its up-to-date information. However, this did not entirely eliminate the inconsistencies and redundancy already present in the system. Indeed, in some instances this "solution" may have exacerbated this situation. It was for these reasons that Excel Industries decided to go the route of a *paperless* or as near possible paperless solution. The project would seek to develop suitable methods for the scanning and uploading of delivery dockets into the Document Management System (DMS).

2.2 Methodology and Approach

2.2.1 Business Process Re-engineering

The business issues addressed included; the degree of business integration, back office procedures, payment processes and methods and customer support.

The methodology employed included the augmentation and adaptation of proprietary business software and the application of such structured systems analysis techniques as; data modelling, decomposition and abstraction.

Each delivery docket has its own unique docket number (SDN). This SDN in conjunction with the corresponding invoice number (SIN) would be used as the basis for achieving the required level of integration.

2.2.2 Development of Imaging System

This required the adaptation of commercially available scanning equipment to enhance the imaging such that draft quality printing could be employed. This required the active participation of a major player in the imaging business as an active stakeholder.

The technical issues that emerged included: scanning and optical character recognition, data compression, storage, updating and retrieval; and modifications to the existing network topology.

The methodologies employed included prototyping commercially available scanning applications and equipment in order to capture draft quality printing; and programming languages, principally XML. The former acts as the input medium to the repository for data storage. The latter extracts specific delivery docket/invoice data and appends it to the corresponding scanned image, which is indexed using standardized document storage formats.

The majority of order/delivery docket printing systems are carried in draft multi-page form. Such poor quality printing can lead to significant errors due to misinterpretation which undermines the reliability and makes existing systems unsuitable for tracking and data entry purposes.

Thus there was a need to develop a system capable of reading poor quality bar codes to a very high degree of accuracy.

2.3 Business System Overview

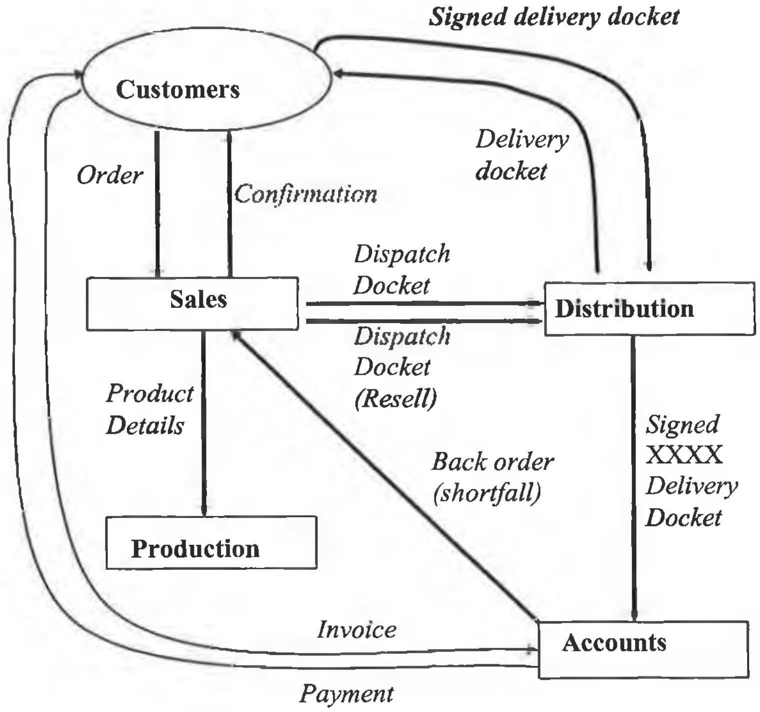


Figure 2.1 Interfacing with Customer Base

Figure 2.1 shows the manner in which Excel Industries interfaced with its customer base. This process generated a large amount of paper-based transactions. Such a system is necessary for the purposes of stock control, auditing, query handling and production scheduling. Transactions can range in financial terms from large sums to very small amounts which need not necessarily reflect the amount of time and effort required to handle a specific transaction and the subsequent customer support it generates.

Thus the cost of supporting some customers could be out of proportion to the amount of actual business activity. Allowance also has to be made for troublesome customers who use queries and other techniques e.g. requests for second proof of delivery dockets, to extend credit facilities and affect the cash flow of the supplier.

2.4 Accounting Function

2.4.1 Typical Query

The scenario showing how a typical query arises is outlined in diagram 2.2. Servicing such requests was carried out by the accounting function in addition to the normal activities. It leads to considerable ineffective and inefficient use of the accounting staff. If the paper chase could be reduced and possibly removed altogether then this would lead to significant savings for the business.

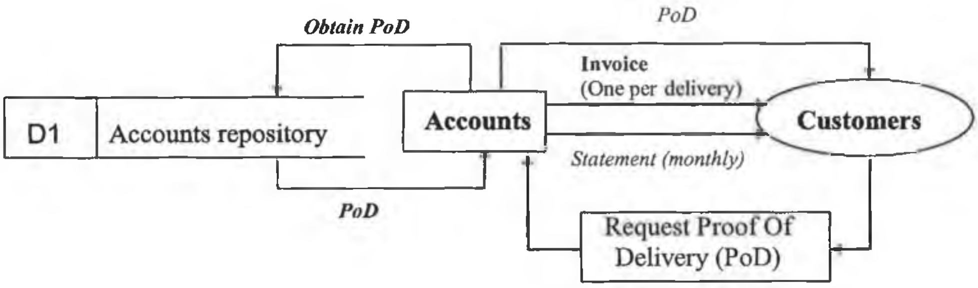


Figure 2.2 Typical Query

2.4.2 Servicing a customer query using the original paper-based system

Figure 2.3 on the following page illustrates the manner in which customer queries were handled by the manual system. On receipt of a query the various paper stores had to be accessed by a member of staff. This included extracting and collating invoices; delivery dockets, letterheads and faxing the entire bundle to the customer in question. This also entailed the temporary storage of this bundle while the query was in dispute. Because a customer might have a number of transactions extant at any one time and these transactions individually might contain a number of items queries could multiply very quickly. This then resulted in considerable demands on the accounting function in terms of time and effort. It also impacted on the cash flow for the company because final payment would not be paid while the transaction was in dispute.

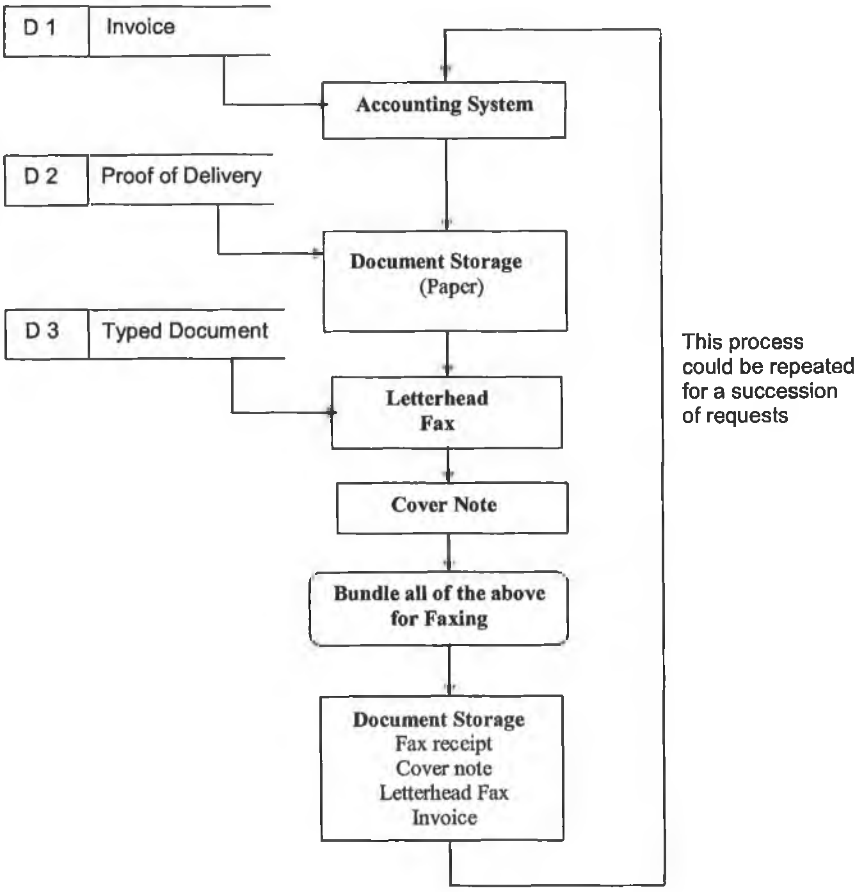


Figure 2.3 Servicing a customer query using original paper-based system

Chapter 3 Selecting an Appropriate Development Model

3 Selecting An Appropriate Development Model

3.1 Introduction

This chapter contains short descriptions of the generally accepted development paradigms. It attempts to locate these models within the wider context of problem solving and summarise why a hybrid model, combining an evolutionary approach with prototyping, was selected.

3.2 Problem Solving

Software development can be characterized as a problem solving loop of four distinct stages:

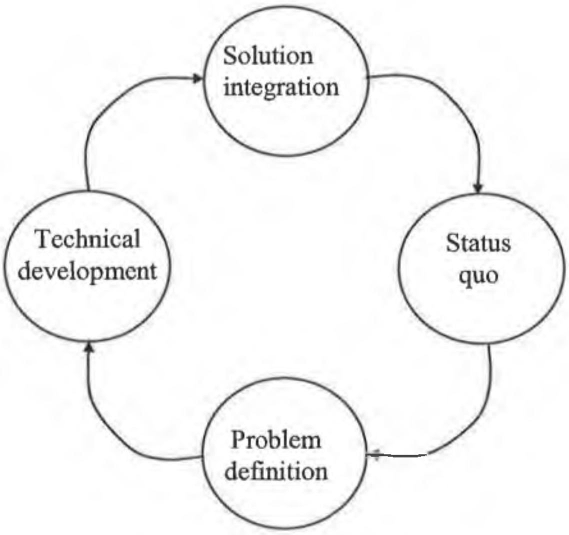


Figure 3.1 Problem solving loop

- *Status quo*: this "represents the current state of affairs" [RAC95]
- *Problem definition*: this identifies the specific problem to be solved
- *Technical development*: solves the problem through the application of some technology,
- *Solution integration*: delivers the results (e.g., documents, programs, data, new business function, new product) to those who requested the solution in the first place.

A process model for software engineering is chosen based on the nature of the project and application, the methods and tools to be used, and the controls and deliverables that are required [PRESS97]. The model is an abstract representation of a process.

Some of the most common models extant at present are briefly detailed

3.3 Typical Process Models [SOMM2001]

- *Waterfall Model (also Linear sequential)*

This represents the process activities as separate process phases e.g. requirements specification, software design, implementation, testing, etc. Once a stage is defined it is signed off and development proceeds to the next stage.

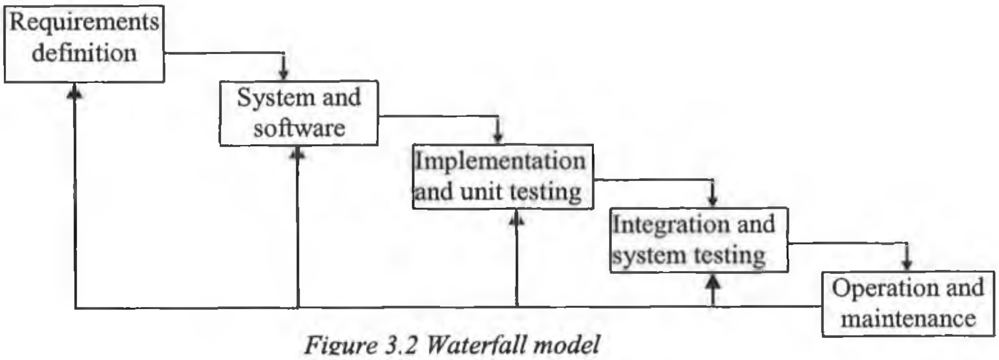

Requirements definition
System and software
Implementation and unit testing
Integration and system testing
Operation and maintenance

Figure 3.2 Waterfall model

- *Evolutionary development*

Fast development of an initial system from abstract specifications which is then refined in conjunction with the client to produce the final product. The activities of specification, development and validation are effectively interwoven.

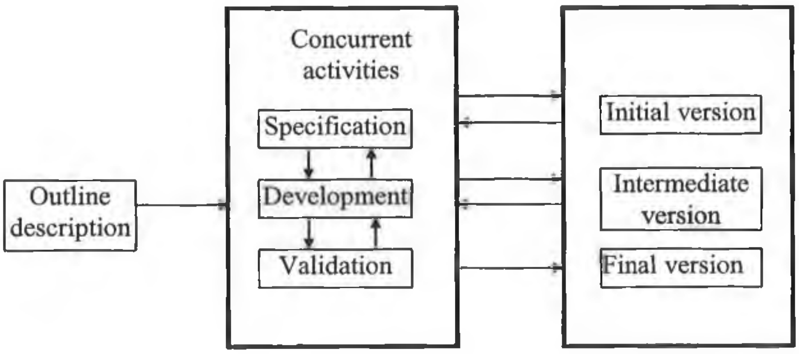


Figure 3.3 Evolutionary Development

- *Formal systems development*

Requires the production of a mathematical system specification and the use of mathematical methods to transform the specification into a program. Verification of conformance is determined by mathematical argument.

- *Reuse-based development*

  This development process focuses on integration of existing reusable components into a system rather than developing them from scratch.

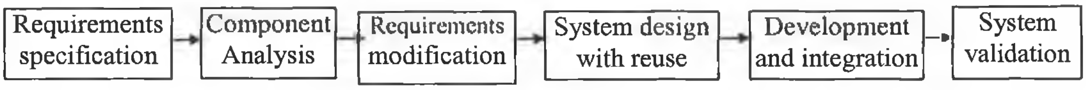

Requirements specification
Component Analysis
Requirements modification
System design with reuse
Development and integration
System validation

Figure 3.4 Development based on Reuse

Because of the cross-talk that occurs within and across stages activities are rarely pigeonholed as neatly as described above. Consequently some or all of the stages may coexist simultaneously. Each process model represents a process from a particular perspective so only provides partial information about that process. For many systems there is no single process that is used and a number of development paradigms may be implemented for different parts of a project thus giving rise to hybrid models such as *incremental [Mills et al 80]* and *spiral [Boehm88]* development. The former combines the advantages of the waterfall and evolutionary approaches while the latter, as the name implies, treats the process as an expanding spiral that ultimately achieves its goal.

The waterfall and formal paradigms were considered inappropriate for the project in hand. The former because the non-sequential nature of the project and the latter because of the unsuitability of the model to express the specification in a mathematical form. Although the ultimate goal was clear, "*move to a paperless replication of the existing paper-based system*", the requirements to achieve this were poorly defined. Thus it was deemed that a hybrid model using an evolutionary approach combined with prototyping would be adopted.

Discussions with Excel management elicited the following narrower but clearer objective:

"Excel Industries required the implementation of a Document Management System (DMS) to allow them to improve their response to customer account queries. This system would allow them to collect payments faster and reduce their Debtors' Days Outstanding (DDO) figures and improve productivity."

Thus the first item on the agenda was an analysis of the existing system.

Chapter 4: Requirements Analysis

4 Requirements Analysis

4.1 Introduction

Excel Industries require the implementation of a Document Management System (DMS) to allow them to improve their response to customer account queries. This system would allow them to collect payments faster and reduce their Debtors' Days Outstanding (DDO) figures.

There are five types of documents that need to be managed:

- Sales Dockets
- Invoices
- Credit Notes
- Delivery Dockets
- Customer Correspondence

The latter is split into those that originate from Excel and those that originate from the customer. The first three documents are produced by the Exchequer Enterprise accounting package used by Excel Industries. The two correspondence types may manifest themselves as hard copy or in electronic format such as email or word processing format. The electronic documents are stored within the DMS in their original format. The hard copy documents are scanned into the system.

On receipt of a customer query Excel staff should be able to enter the relevant details of the document into the Document Management System and view that document and all of the other documents relating to the query. The system should permit copies of stored documents to be sent to Excel's customers via fax or email.

Market research revealed that a number of companies - SoftCo, Adest, FileNet, ADOS, Scanfile – would be capable of implementing a document management system. ADOS and Scanfile were considered to offer the best potential for the task in hand in terms of scalability and support. The others were rejected for the following reasons:

- ❑ Inappropriate scale of the product for the task in hand
- ❑ The required level of maturity of the product
- ❑ Poor maintenance history in regard to these types of projects.

Negotiations took place with both companies and Excel Industries chose ADOS as its preferred option. The bases of this decision were commercial and technical. Suffice to say that the commercial factors are beyond the scope of this project for reasons of commercial sensitivity. In keeping with the adopted development paradigm a programme of prototyping and incremental development was initiated to clarify the requirements for the system.

4.2 Prototyping

Figure 4.1 shows the sequence chosen for the prototyping process.

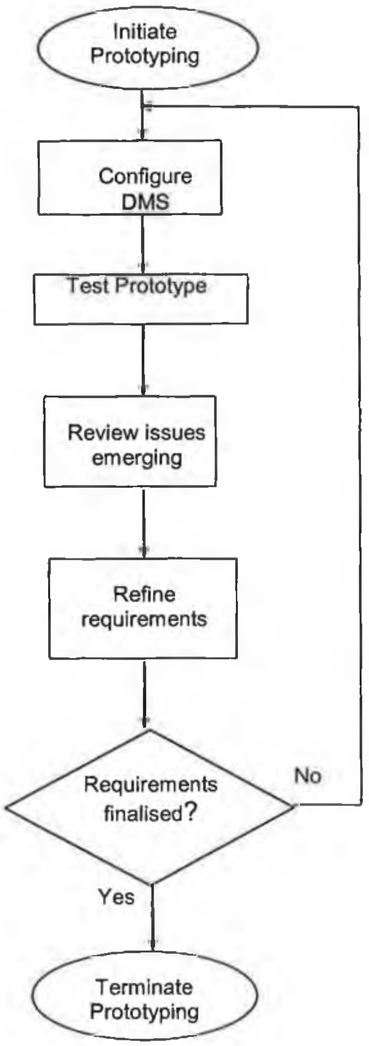


Figure 4.1 Flowchart for Prototyping

4.2.1 Version 1

Key features:

- ADOS platform
- Documents scanned and saved as bitmaps
- Manual input of data

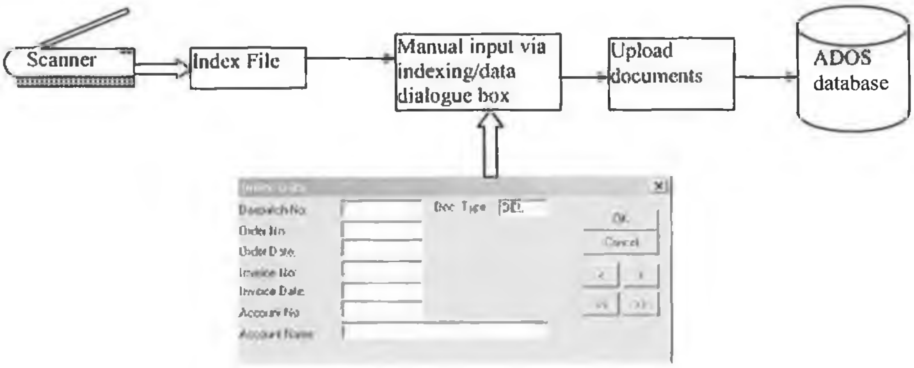


Figure 4.2 Configuration of Prototype 1

The ADOS archive and retrieval system processes, indexes, stores, and manages the scanned documents. The documents are processed and saved locally or remotely. The system stores the archived information in an SQL database. The ADOS system was configured to permit manual input of relevant data. The system as shown in the figure above required the use of a dedicated data inputter. A data inputting exercise was carried out over a five-day period to establish the workload that such an endeavour would entail.

Five different people within the company with varying levels of expertise were given the task of transferring the data from the paper-based form to the ADOS system. The average number of completed entries was about 200. The most experienced never got above 240 or so without a concomitant loss in accuracy. Simple errors such as an 'o' in place of a zero when inputting the unique docket number could result in the associated data being useless. The accuracy achieved never exceeded 70%.

The average time per docket was 2 minutes, a time established as the optimum interval in order to minimize input errors. (200 x 2 minutes = 400 minutes = more than 6 and a half hours i.e. an entire days work.) Thus the idea of a dedicated data inputter was discounted. In addition, because dockets for inputting arrive on ad hoc and infrequent basis it was not possible to schedule this person's work efficiently.

Scanning the entire docket as a bitmap yielded very large file formats with the consequent problems of slower network speed and the requirement for large amounts of data storage. In addition, customers who had agreed to partake in the trialling of the system were unhappy with the large data volumes traversing their systems e.g. fax machines would be tied to Excel for long periods while information was being transmitted. In an attempt to reduce the size of the files generated a decision was made to change the company logo from colour to black and white. This decision yielded some reduction but the effect was not sufficient to solve the underlying problem.

In summary the issues emerging were:

- Large file sizes
- Low transfer speeds
- Implications for the network
- Poor performance
  - Accuracy of stored data
  - Excessive time required

The difficulties encountered as outlined above led to the decision to reject the ADOS platform and move to the Scanfile option. The implications of large volumes of data clogging up the network resulted in a decision to upgrade the network in order to future-proof against whatever system might evolve.

4.2.2 Version 2

Key Features

- Scanfile platform
- Documents scanned and saved to tailored file structure
- Specific fields read from docket e.g. SDN
- SDN chosen as unique identifier for scanned docket
- Redesign of delivery docket from landscape to portrait
- Images in TIFF

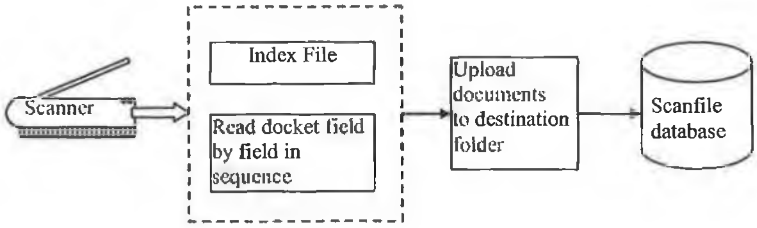

Scanner
Index File
Read docket field by field in sequence
Upload documents to destination folder
Scanfile database

Figure 4.3 Configuration of Prototype 2

The scanned documents are stored in folders created for specific types of documents e.g. invoices. These are saved locally or on the network. The folders contain the scanned documents and the associated indexing information. A folder directory is located in the root directory of the local drive.

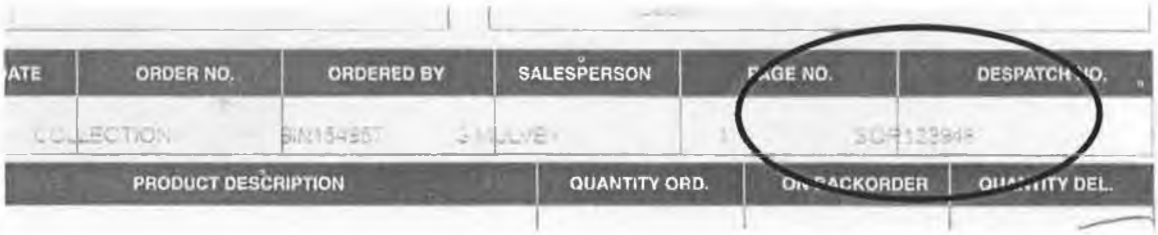
The Scanfile application appeared well-suited to the major needs of the project. The issue of the large file sizes occurring in version 1 was addressed by migrating to Tagged Image File Format (TIFF) compressed output from the scanner.

However, the original form of the docket still presented difficulties. The optical character recognition (OCR) that lies at the heart of the Scanfile application encountered difficulties when processing fields on the docket which contained low quality printing e.g. 'O' mistaken for zero. (See figure 4.4 below).

Consideration was given to the use of high-grade laser printing. This was rejected because the delivery dockets are in multi-page format for the purpose of proof of delivery and audit trailing. Each copy of the delivery docket has to be signed. Laser printing would mean the production of separate copies, which would have to be collated and then signed individually on receipt of goods by the customer. The multi-page, carbon-backed format obviates the need for multiple printing of dockets.

A partial solution was arrived at by re-orientating the docket format from landscape to portrait. This permitted the use of additional lines on the docket but past experience predicted that there was no guarantee that these additional lines would not be exceeded on a single docket. Nonetheless this feature was carried through to the next version.

In addition, the Scanfile system was configured to read specific regions of the docket fields. Any deviation from this produced misreadings of the data, particularly the SDN that had been chosen as the unique identifier for retrieval purposes. (See figure 4.4 below). As a consequence of these two complications the overall accuracy only increased to 80%.



Potential misread of 'O' and shift in location of data

ATE	ORDER NO.	ORDERED BY	SALESPERSON	AGE NO.	DESPATCH NO.
COLLECTION		SIN154957 S MULVEY		1	SOR123948

PRODUCT DESCRIPTION	QUANTITY ORD.	ON BACKORDER	QUANTITY DEL.

Figure 4.4 Extract from incorrectly scanned docket

In summary the issues emerging were:

- Use of SDN as unique identifier
- Common default location for key identifier
- Printing
    - Jitter and misalignment of low quality printing source causing misreading of data.
    - Rejection of high quality laser scanning.
    - Inability to cope with multiple copies of same document.
    - Insufficient accuracy of retrieval due to misinterpretation of data

In spite of the above deficiencies it was felt that the Scanfile platform could be configured to overcome the problems and be tailored to the needs of the system. However, this avenue was not pursued when, for the sound commercial reason of cost, ADOS re-emerged as a potential provider. Additionally, ADOS was proposing the application of a *Kofax* module. This module permits photographing, dynamic enhancement and scanning of the selected portion separate to the scanning of the delivery docket itself.

4.2.3 Version 3

Key Features

- ADOS Platform
- Kofax Module introduced
- SDN as unique identifier for scanned docket but now in barcode format
- Manual input of SDN in the event of failure to read bar code.

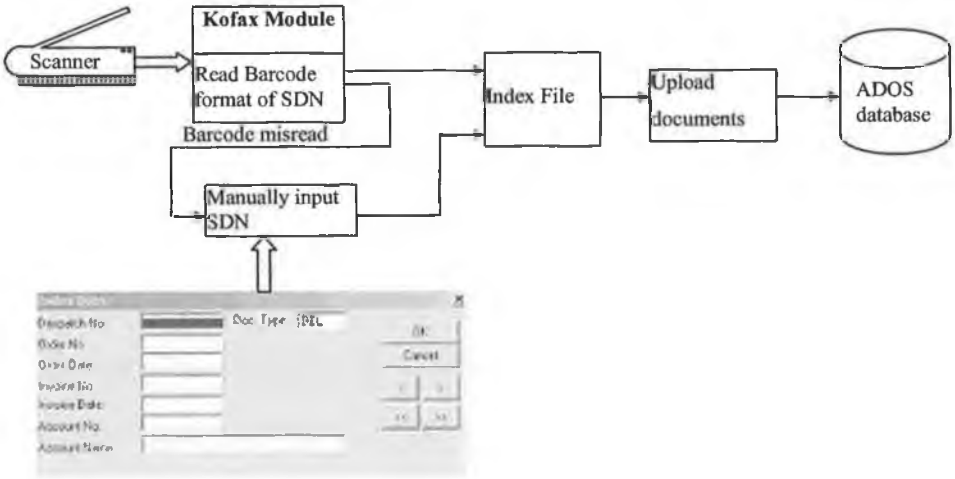
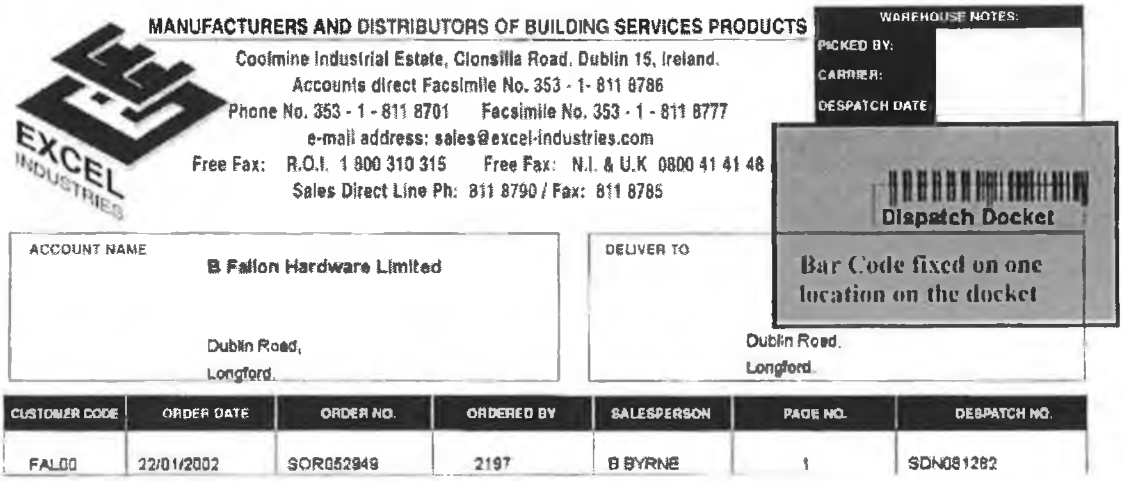


Figure 4.5 Configuration of Prototype 3

The new prototype would use the Kofax module to target a specific area on the docket reserved for bar coding. Ancillary data would be inputted manually to obviate the problems with scanned numbers and letters as experienced with version 2. (See diagram below)



EXCEL INDUSTRIES

MANUFACTURERS AND DISTRIBUTORS OF BUILDING SERVICES PRODUCTS

Coolmine Industrial Estate, Clonsilla Road, Dublin 15, Ireland.
Accounts direct Facsimile No. 353 - 1- 811 8786
Phone No. 353 - 1 - 811 8701 Facsimile No. 353 - 1 - 811 8777
e-mail address: sales@excel-industries.com
Free Fax: R.O.I. 1 800 310 315 Free Fax: N.I. & U.K 0800 41 41 48
Sales Direct Line Ph: 811 8790 / Fax: 811 8785

WAREHOUSE NOTES:
PICKED BY:
CARRIER:
DESPATCH DATE

Dispatch Docket

Bar Code fixed on one location on the docket

ACCOUNT NAME
B Fallon Hardware Limited
Dublin Road,
Longford.

DELIVER TO
Dublin Road,
Longford.

CUSTOMER CODE	ORDER DATE	ORDER NO.	ORDERED BY	SALESPERSON	PAGE NO.	DESPATCH NO.
FAL00	22/01/2002	SOR052949	2197	B BYRNE	1	SDN081282

Figure 4.6 Fixing default location for bar code

The ADOS application was configured to read the bar code and index the document. Manual input of SDN to be provided in the event of failure to read bar code. The European Assistance Network (EAN) Numbering System 13 was chosen as the bar code symbology. The combination of Kofax module and bar code proved to be a significant development in that reading accuracy increased to 90%.

Whilst this improvement was welcomed it still fell short of what was acceptable. This issue needed to be addressed. The problem stemmed from the low-level print quality of the bar code. High quality printing was not an option for the reasons previously stated. Thus the barcode format would need to be changed to one that was relatively immune to gain or loss across the symbol. EAN 128 offered such an alternative.

A series of test scans were carried out on low quality bar codes to compare the relative efficacies of EAN 13 and EAN 128. EAN 128 proved to be the more robust, producing accuracies of greater than 98% compared to 91 % for EAN 13. Thus EAN 128 was selected as the bar code format going forward. (See section 5.1 for further details on bar coding).

A brief outline of the EAN requirements for bar code verification is included in Appendix 1. This outlines how verification can be achieved. However, this does not address the fundamental task that had to be undertaken i.e. the necessity to achieve a bar code quality sufficient for scanning *and* simultaneous verification

The model realised by version 3 still required the manual inputting of data to facilitate searches of the DMS by criteria other than the SDN. What had not been appreciated heretofore was that this information was already available to the company albeit within another application namely, the Exchequer accounting package. It was decided to integrate the two systems.

In summary the issues emerging were:

- SDN now in bar code form
- Selection of bar code format
- Proposal to merge with existing accounting software

The requirements for the project had evolved into the following set of statements:

To provide a fully working document management system that is capable of meeting the following requirements:

- All new and historic delivery notes to be scanned into the system and indexed.
- Extraction of Sales Invoices and Credit Notes from the Exchequer software during a daily run and the importation of these into the DMS.
- All supplier correspondence to be stored in the DMS
- All data extracted from the Exchequer accounting package, to be available from the DMS.
- Upgrading of the network to facilitate the proposed system.

Chapter 5: System Development

Section **Topic**

5 System Development

5.1 Final Version

The outcome of the prototyping process was incorporated into version 4 which was then scaled up to a full working system. The key features of the new model are as listed below and figure shows the configuration of the new version.

- Version 3 accepted as working model for DMS
- Integration of DMS with accounting package
- BASDA selected as schema for XML
- XML/XSL coding required for integration
- Adoption of EAN 128 as bar code format

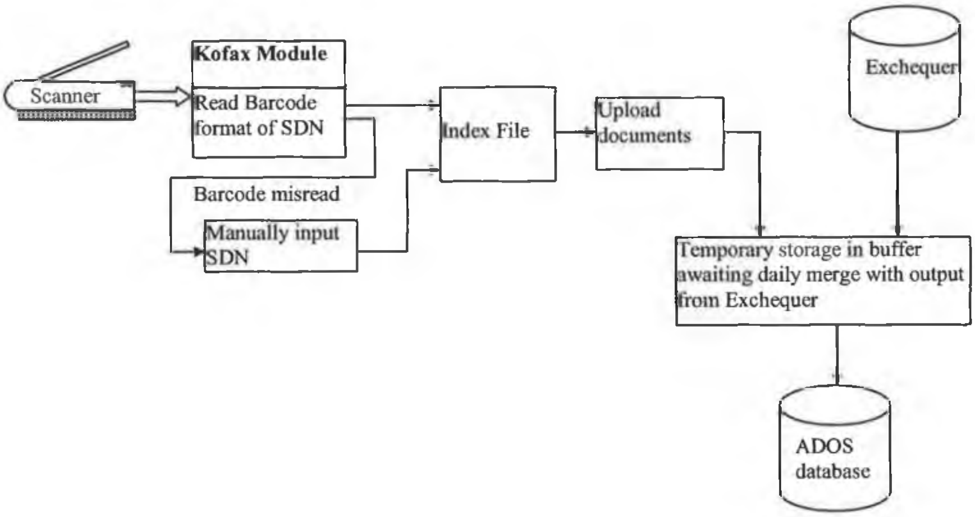

Scanner
Kofax Module
Read Barcode format of SDN
Barcode misread
Manually input SDN
Index File
Upload documents
Exchequer
Temporary storage in buffer awaiting daily merge with output from Exchequer
ADOS database

Figure 5.1 Configuration of Version 4 (Final Version)

Exchequer were requested and agreed to permit the required level of integration provided that, for reasons of security and version control, the necessary XML code be developed by Excel Industries under their supervision. The structure for the integration is shown in figure5.2.

The scanned delivery dockets are temporarily stored in a buffer according to the SDN derived from the scanned bar code. Posted invoices are outputted from the accounting system on the basis of business needs and put into COLD (Computer Output to Laser Disk)storage. [Despite its name COLD is not limited to the laser disk medium]. COLD software indexes the outputted files for further processing.

Every night an SQL function interrogates the two storage systems and ties the delivery docket to its corresponding data. This process removes the need to manually input the data as was the case with previous versions.

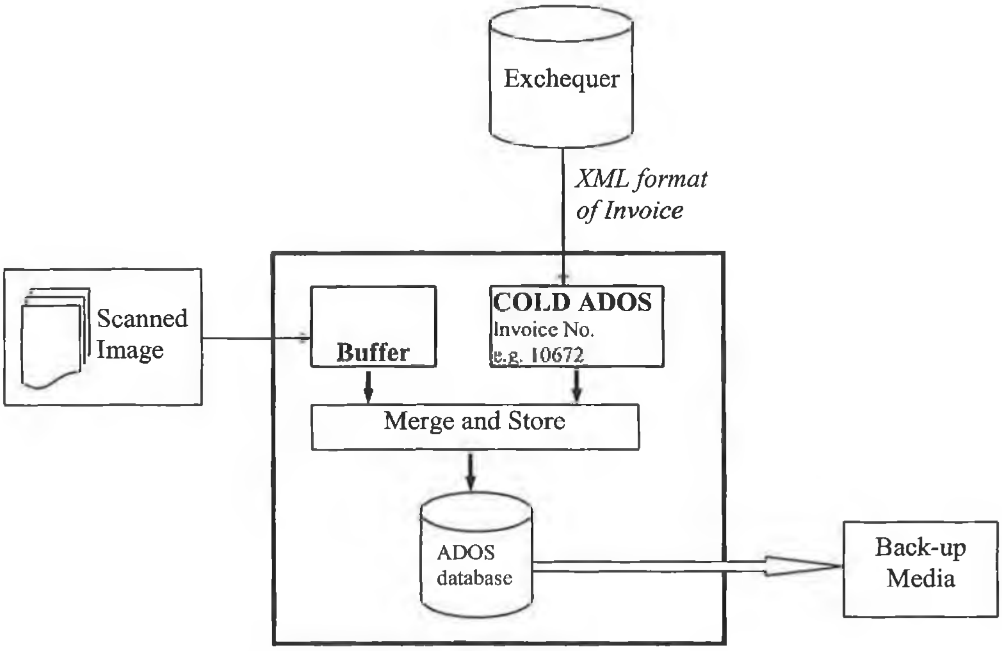


Figure5.2 Schematic of Merging Process

5.2 Bar Code Generation

5.2.1 Introduction

A rigorous approach needs to be undertaken when adapting bar code symbologies for the purposes of identification. Despite this problems still derive from the quality of the bar codes produced. The impact of poor quality printing is dealt with in chapter 4. In the context of this project the implications for the business processes and customer confidence of misinterpreted bar codes were very significant. The following details the underlying structure of the chosen bar code format.

5.2.2 European Assistance Network (EAN) Numbering System

EAN International is the body that specifies and administers standard numbers for identification purposes. Each number is considered to be a means of identification and not a means of classification. Individual digits in the numbers are not vested with any particular significance and do not convey information about the item or its location. The numbers are formed from parts to ensure uniqueness and easy management so that there is no danger of ambiguity.

EAN Prefix	Structure determined by the relevant Numbering Organization	Check Digit
P_1 P_2 P_3	N N N N N N N N N	C

Figure 5.3 Typical EAN format

The prefix refers to the national EAN numbering organization and C is a calculated check digit.

Bar code symbols EAN-13 and EAN-128 comply with the EAN-13 number structure. The following selection process dictates the choice of format.

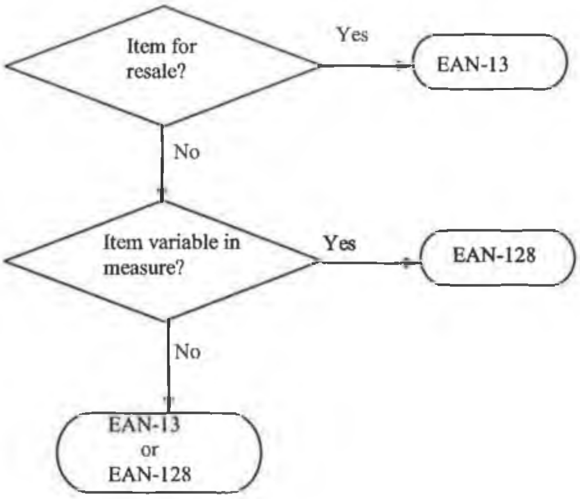


Figure 5.4 Bar Code Selection

Check Digit Algorithm

13-Digit Article Numbers

The check digit is calculated by a modulo-10 algorithm from all the other digits in the number by means of the following steps.

Step 1: Start with the digit on the right of the number, (exclude the check digit) and sum all the alternate digit values, reading from right to left.

Step 2: Multiply the result of step 1 by 3.

Step 3: Sum all the remaining digit values.

Step 4: Add the result of step 2 to the result of step 3.

Step 5: The check digit is the smallest number which when added to the result of step 4 produces a multiple of 10.

	Digit Position												
	13	12	11	10	9	8	7	6	5	4	3	2	1
Number	5	0	2	6	4	5	1	7	3	8	8	5	C
Step 1		0	+	6	+	5	+	7	+	8	+	5	= 31
Step 2	Multiply by 3												= 93
Step 3	5	+	2	+	4	+	1	+	3	+	8		= 23
Step 4	Add Step 2 and Step 3												= 116
Step 5	Check Digit is 120 minus 116												= 4

Table 5.1 Calculating the check digit for the article number 502645173885 C:

The process reveals that the check digit is 4.

5.2.3 EAN-128

The EAN number system is capable of being represented by machine-readable bar codes. Earlier versions were designed to optimise efficiency of retail checkouts. EAN-128 has been designed to provide additional encoding and is not intended for point-of-sale purposes.

5.2.4 Symbology Identifiers

All scanning equipment has the ability to recognize the symbology of the bar code that has been scanned but some scanners have the optional feature of being able to transmit a symbology identifier comprising a three-character data string. For EAN -128 symbols this is **]cm**, where:

] is the *Flag Character* (ASCII value of 93) - this indicates that the two
characters
following are symbol identifier characters
c is the *Code Character* and this indicates the bar code symbology, which for EAN 128 is an *upper case* 'C';
m is the *Modifier Character* which indicates the mode in which the symbology is used.

5.2.5 EAN-128 symbols have the following characteristics:

- Rectangular shape made up of a series of parallel bars of uniform width, light or dark
- Characters are made up of 11 modules, light or dark, except the stop character which is made up of 13 modules;
- The character modules are grouped into bars, with each character represented by three light bars and three dark bars, except the stop character which is made up of three light bars and four dark bars
- The symbol has a double character start pattern consisting of the appropriate start character immediately followed by the **Function 1 (Fnc.l)** character;
- The symbol always incorporates a symbol check character which is *not* part of the data and is additional to any check digits used in the data
- The symbol has a light margin to the left and right and is designed to be read bi-directionally by the scanner
- The symbol size depends on the number of characters encoded

5.2.6 Character Representation

EAN-128 symbology has three character sets:

- Set A - includes all standard upper case alpha-numeric characters plus some control characters
- Set B - represents all standard upper case and lower case characters plus special characters;
- Set C - includes the set of 100 digit pairs from 00 through 99 as well as special characters.

The start character determines which set is being used. The stop character terminates the symbol.

	D	L	D	L	D	L
Start (A)	2	1	1	4	1	2
Start (B)	2	1	1	2	1	4
Start (C)	2	1	1	2	3	2

	D	L	D	L	D	L	D
Stop	2	3	3	1	1	1	2

Table 5.2 The character sets for the start and stop characters.

D = Dark bar L = Light bar

5.2.7 Limitations in Symbol Size

The size of the EAN-128 bar code symbol depends on three factors:

- The magnification factor
- The number of data characters encoded
- The number of non-numeric data characters encoded.

The formula for calculating the symbol width W (in millimetres), including light margins, is:

$$W = (11n + 22) \times M$$

Where n is the number of symbol characters. The symbol is limited to a maximum width of 165mm.

5.2.8 Format of the EAN-128 Bar Code

The EAN-128 bar code is made up as follows (reading the bar code from left to right):

- A light margin
- A start character
- A function 1 character
- An amount of data
- A symbol check character
- A stop character
- A light margin.

5.2.9 Symbol Check Character

A symbol check character is included in the EAN-128 bar code symbol. It appears immediately preceding the stop character. It is calculated by using the modulo 103 algorithm described below.

The modulo 103 algorithm makes use of *all* the characters in the symbol except the stop character. The algorithm comprises the following steps.

Step 1 Weight the characters in the symbol by multiplying the value of the start character by 1, the value of the function 1 character by 1, the value of the first data character by 2 and each remaining data *and* auxiliary character with ascending weights 3, 4, 5 and so on.

Step 2: Sum all the weighted values.

Step 3: Divide the result of step 2 by 103.

Step 4: The remainder from the division in step 3 is the value that corresponds to the value of the check character in the EAN table.

Example The article number 5012345678900 and date code 1st January 2004 are represented in EAN-128 by the characters:

Start C, Function 1, 010501234567890015040101, Check Character, Stop

The value of the check character is calculated as follows:

Step 1

Character	**Weighting**	**Value**
Start C (105)	1	105
Function 1 (102)	1	102
01	2	2
05	3	15
01	4	4
23	5	115
45	6	960
67	7	469
80	8	712
00	9	0
15	10	150
04	11	44
01	12	12
01	13	13

Step 2 Total = 2703

Step 3: 2703/103 = 26 remainder 25

The value of the check character is 25.

The diagram below shows the comparative physical size for bar code symbols generated using EAN-13 and EAN-128 respectively. EAN-128 was adopted as the format for the new document processing system despite its larger size. (This had implication for its siting on the dockets). Both symbols were put through a number of batch tests in collaboration with Canon-ADOS using the modified scanning equipment. The results of these tests proved conclusively that EAN-128 was more reliable and accurate for the requirements of the proposed system. This explains why detailed reference is made to EAN-128 and only superficial reference to EAN-13.

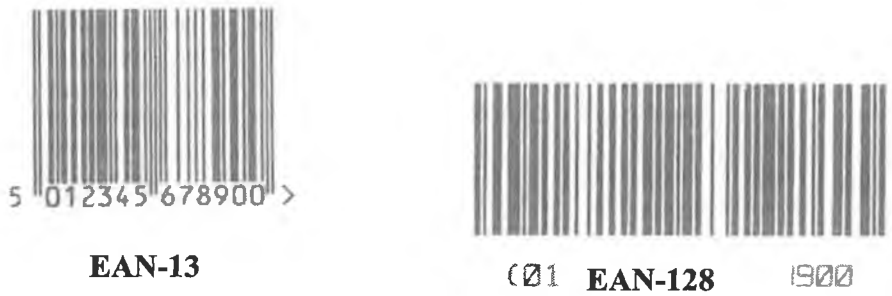


Figure 5.5 Bar code sizes

5.3 XML Schemas

The XML schema is a mechanism for defining XML structure and what a given set of one or more XML documents can look like.. It identifies a set of components in an XML document and provides the rules for the correct combination of these components in terms of their order, content and the attributes they contain. The schema defines; the *elements* and *attributes* that appear in a document, the *default values* for attributes, the number of *child elements* and the *sequence* in which child elements can appear. In addition the schema supports data types, is extensible and is written in XML.

The XML Schema language, developed by the W3C, is written in XML and permits the definition of both global elements and local elements. The former must be used in the same way throughout the XML document whilst the latter can have a particular meaning in a particular context. XML Schemas also contain a system of *datatypes* that specify, for example, that one element should contain an integer and that another should contain a string, etc.

A schema is divided into types of content:

- *Simple Type:* Elements that contain only text.

```
<ElementType name="EarliestAcceptableDate" content="textOnly"
```

- *Complex Type:* Elements that contain other elements or contain attributes.

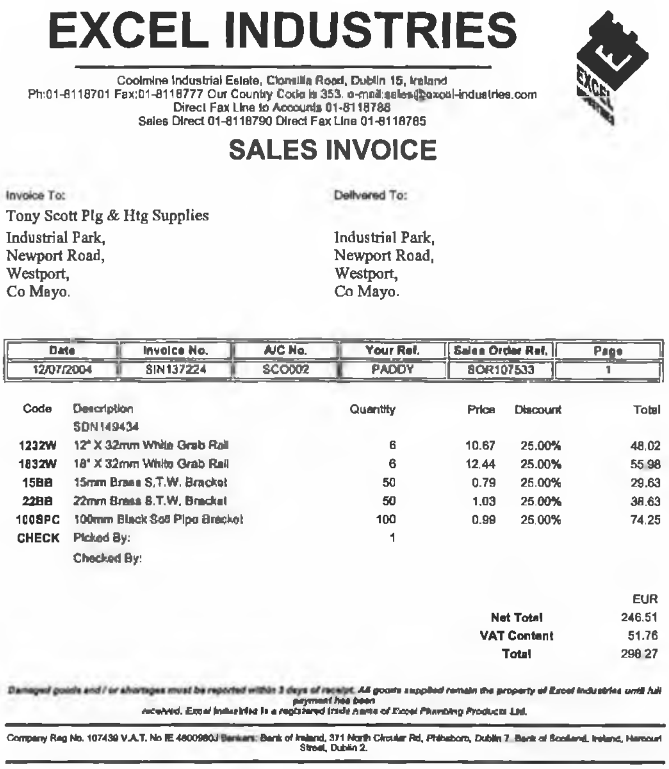
```
<ElementType name="Invoice" content="eltOnly" order="seq">
          <element type="InvoiceHead"/>
          <element type="InvoiceReferences"/>
          <element type="InvoiceDate"/>
          <element type="TypeOfSupply" minOccurs="0" maxOccurs="1"/>
          <element type="TaxPointDate" minOccurs="0" maxOccurs="1"/>
          <element type="Supplier"/>
          <element type="Buyer          " minOccurs="0" maxOccurs="1"/>
          <element type="Delivery" minOccurs="0" maxOccurs="*"/>
          <element type="InvoiceTo"/>
          <element type="InvoiceLine" minOccurs="1" maxOccurs="*"/>
          <element type="PercentDiscount" minOccurs="0" maxOccurs="*"/>
          <element type="AmountDiscount          " minOccurs="0" maxOccurs="*"/>
          <element type="SpecialInstructions" minOccurs="0" maxOccurs="*"/>
          <element type="Narrative" minOccurs="0" maxOccurs="*"/>
          <element type="Settlement" minOccurs="0" maxOccurs="1"/>
          <element type="TaxSubTotal" minOccurs="1" maxOccurs="*"/>
          <element type="InvoiceTotal"/>
</ElementType>
```

The BASDA schema was used as a template for the XML coding. The two samples above are taken from the BASDA Schema and entire schema can be seen in “APPX 2.

Schema components that are declared at the top level of a schema are considered to be *globally declared.* These are available to be used throughout the rest of the schema. Globally declared elements only determine what that element will look like and not where it appears. Complex type elements can either reference existing globally declared elements or they can declare and define new elements. These are referred to as locally declared types. The position in which these elements are defined determine where they will appear in the XML document.

5.3.1 Schema Map for Sample Invoice/Credit Note

The diagram on the subsequent pages outline the schema map for the sample invoice shown and this is derived from the BASDA format. This indicates the elements and attributes that are contained within in the particular invoice type.



EXCEL INDUSTRIES

Coolmine Industrial Estate, Clonsilla Road, Dublin 15, Ireland
Ph:01-8118701 Fax:01-8118777 Our Country Code is 353. e-mail:sales@excel-industries.com
Direct Fax Line to Accounts 01-8118788
Sales Direct 01-8118790 Direct Fax Line 01-8118785

SALES INVOICE

Invoice To:
Tony Scott Plg & Htg Supplies
Industrial Park,
Newport Road,
Westport,
Co Mayo.

Delivered To:

Industrial Park,
Newport Road,
Westport,
Co Mayo.

Date	Invoice No.	A/C No.	Your Ref.	Sales Order Ref.	Page
12/07/2004	SIN137224	SCO002	PADDY	SOR107533	1

Code	Description	Quantity	Price	Discount	Total
	SDN149434				
1232W	12" X 32mm White Grab Rail	6	10.67	25.00%	48.02
1832W	18" X 32mm White Grab Rail	6	12.44	25.00%	55.98
15BB	15mm Brass S.T.W. Bracket	50	0.79	25.00%	29.63
22BB	22mm Brass S.T.W. Bracket	50	1.03	25.00%	38.63
100SPC	100mm Black Soil Pipe Bracket	100	0.99	25.00%	74.25
CHECK	Picked By:	1			
	Checked By:				

	EUR
Net Total	246.51
VAT Content	51.76
Total	298 27

Damaged goods and / or shortages must be reported within 3 days of receipt. All goods supplied remain the property of Excel Industries until full payment has been received. Excel Industries is a registered trade name of Excel Plumbing Products Ltd.

Company Reg No. 107439 V.A.T. No IE 4600980J Bankers: Bank of Ireland, 371 North Circular Rd, Phibsboro, Dublin 7. Bank of Scotland, Ireland, Harcourt Street, Dublin 2.

Figure 5.6 Sample Invoice

Invoice	**Invoice Head**	**Schema**	**Version**	
		Stylesheet	**StylesheetOwner**	
			StylesheetName	
			Version	
			StylesheetType	
		Parameters	**Language**	
			DecimalSeparator	
			Precision	
		OriginatingSoftware	**SoftwareManufacturer**	
			SoftwareProduct	
			SoftwareVersion	
		InvoiceType	***Code***	
		InvoiceCurrency	**Currency**	**Code**
		Intrastat	CommodityCode	
			CommodityDescription	
			NatureOfTransaction	***Codelist***
			SupplementaryUnits	
			CountryOfDestination	
			ModeOfTransport	***Codelist***
		Checksum		
	Invoice References	BuyersOrderNumber	***Preserve***	
		SuppliersInvoiceNumber	***Preserve***	
	Invoice Date			
	Suplier	SupplierReferences	BuyersCodeForSupplier	
			TaxNumber	
		Party		
		Address	AddressLine	
			AddressLine	
			AddressLine	
			AddressLine	
	Buyer	BuyerReferences	SuppliersCodeForBuyer	
		Party		
		Address	AddressLine	
			AddressLine	
			AddressLine	
			AddressLine	
		Contact	Name	
			DDI	
			Fax	

Invoice	Delivery	DeliverTo	**Party**		
			Address	AddressLine	
				AddressLine	
				AddressLine	
				AddressLine	
		LatestAcceptableDate			
	InvoiceTo	InvoiceToReferences	SuppliersCodeForInvoiceTo		
		Party			
		Address	AddressLine		
			AddressLine		
			AddressLine		
			AddressLine		
		Contact	Name		
			DDI		
			Fax		
	InvoiceLine	***Action***			
		TypeCode			
		TypeDescription			
		LineNumber	***Preserve***		
		InvoiceLineReferences	CostCentre		
			GeneralLedgerCode		
			OrderLineNumber		
			BuyersOrderLineReference	***Preserve***	
		Product	SuppliersProductCode		
			ConsumerUnitCode		
			Description		
			Properties	Weight	*UOM Description*
					UOMCode
		Quantity	***UOMDescription***		
			Packsize		
			Amount		
		Price	***UOMDescription***		
			UnitPrice		
		PercentDiscount	Type	***Code***	
			Percentage		
		LineTax	**TaxRate**	**Code**	
			TaxValue		
		LineTotal			
		Intrastat	CommodityCode		
			CommodityDescription		
			NatureOfTransaction	***Codelist***	
			SupplementaryUnits		
			CountryOfDestination	*Codelist*	
			ModeOfTransport	*Codelist*	
	Settlement	SettlementTerms	PayByDate		
		SettlementDiscount	Percent Discount	Qualifying Terms	**PayByDate**
					DaysFromInvoice
				Percentage	
			Amount Discount	Qualifying Terms	**PayByDate**
					DaysFromInvoice
				Amount	
		TaxSubTotal	**TaxRate**	***Code***	
			NumberOfLinesAtRate		
			TotalValueAtRate		
			SettlementDiscountAtRate		
			TaxableValueAtRate		
			TaxAtRate		
			NetPaymentAtRate		
			GrossPaymentAtRate		
			TaxCurrency	**Currency**	***Code***
		InvoiceTotal	**NumberOfLines**		
			NumberOfTaxRates		
			LineValueTotal		
			TaxableTotol		
			NetPaymentTotal		
			GrossPaymentTotal		

Key to references	**Mandatory Element**	Optional Element	***Mandatory Attribute***	*Optional Attribute*

Figure 5.7 Schema Map for Sample Invoice

5.4 Extensible Markup Language (XML)

XML is a markup language for documents containing *structured information.*

- *Structured information:* Information that contains both content and some indication of what role that content plays.

The content may be words, pictures, etc. and the role of the content may be determined by its place in the document. For example, content in a section heading has a different meaning from content in a footnote. This may mean something different than content in a figure caption or content in a database table, etc.

Almost all documents have some structure. A markup language is a mechanism to identify structures in a document. The XML specification defines a standard way to add markup to documents. The word "document" refers to traditional documents and the vast array of other XML "data formats". The latter can include vector graphics, e-commerce transactions, mathematical equations, object meta-data and server APIs inter alia.

XML differs significantly from HTML. In HTML, both the tag semantics and the tag set are fixed. The World Wide Web Consortium (W3C), in conjunction with browser vendors and the WWW community updates and amends the definition of HTML to allow new tags to keep pace with changing technology and to bring variations in presentation (stylesheets) to the Web. Changes are most often determined by the browser vendors and by their desire for backward compatibility.

XML is a meta-language for describing markup languages and it provides a facility to define tags and the structural relationships between them. Because there is no predefined tag set there cannot be any preconceived semantics, all of the semantics of an XML document will either be defined by the applications that process them or by stylesheets.

XML is an application the Standard Generalized Markup Language (SGML) as defined by ISO 8879. SGML is a vendor-independent standard designed to maintain repositories of structured documentation. Because XML is defined as an application of SGML this means that any system that fully conforms to the SGML will be able to read XML documents. However, using and understanding XML documents does not require that a system be capable of understanding the full extent of SGML. It is feasible to consider XML as a restricted form of SGML. This distinction can lead to subtle differences between documents as understood by XML systems and those same documents as understood by SGML systems.

The diagram describes the typical phases in the life cycle of an XML document. Each stage is essentially independent of each other; the stages are connected only by the XML document itself. The function of the parser is to read the document and to check if the XML content is *well-formed*.

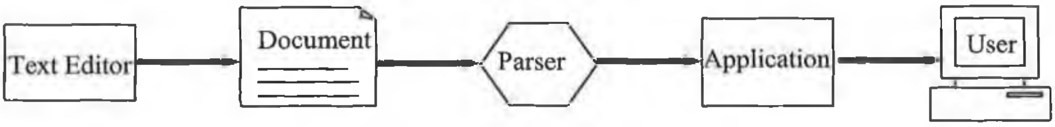


Figure 5.8 Life Cycle of an XML Document

5.4.1 eCommerce

eCommerce speeds up the supply chain and eliminates unnecessary stages. It enables manufacturers to deal directly with re-sellers and consumers and also addresses typical transactions between businesses in an electronic form. The major advantage to business is the elimination of re-keying data into systems.

eCommerce can be divided up into several segments:

- Consumer to business (c2b)
- Business to business (b2b)
- Citizen to government (c2g)
- Business to government (b2g)

5.4.2 eBIS-XML

The Business Application Software Developers Association (BASDA) represents more than 300 of the world's leading applications software developers. It is recognized by United Nations, the European Commission many governments throughout the world. BASDA has developed an electronic business interchange standard in XML known as eBIS-XML. This standard enables the direct exchange of purchase orders and invoices and other business documents between different software packages, via e-mail and the Internet, without the need for EDI (Electronic Data Interchange) middleware or Value Added Networks.

The deployment of eBIS-XML permits interoperable eCommerce between standard software packages by means of the implementation of XML schemas. Messages received by systems, which are not eBIS-XML enabled, can simply be displayed and printed out as documents. Thus a company does not need to know if its supplier or customer is eBIS-XML enabled before it sends an eBIS-XML order or invoice.

5.5 Extensible Style Language (XSL)

Browsers know how to display the predefined HTML tags but browsers do not automatically understand XML tags. For example <table> could mean a HTML table or maybe a piece of furniture. Due to the nature of XML there is no standard way to display an XML document. To display XML documents, it is necessary to have a mechanism to describe how the document should be displayed. One such mechanism is the e**X**tensible **S**tylesheet **L**anguage (XSL).

XSL includes *a transformation* language and a *formatting* language, that can function independently of each other and, each of which is an XML application. The transformation language defines rules for how one XML document is transformed into another XML document.

The ability of the transformation language to move data from one XML representation to another makes it an important component of XML-based electronic commerce, electronic data interchange, metadata exchange, and any application that needs to convert between different XML representations of the same data

Every *well-formed* XML document is a tree i.e. a hierarchical structure composed of connected nodes beginning with a top node called the root node. The root is connected to child nodes, each of which is connected to zero or more children of its own and so on. Each node and its children also form a tree. Elements, attributes, namespaces, processing instructions, and comments are counted as nodes. XSLT processors model an XML document as a tree that contains seven kinds of nodes:

- The root
- Elements
- Text
- Attributes
- Namespaces
- Processing instructions
- Comments

5.5.1 The XSLT Transformation Process

In an XSL transformation, an XSLT processor reads both an XML document and an XSLT style sheet. The processor uses the instructions in the XSLT style sheet to output a new XML document or HTML output.

The steps involved include:

1. Interpret the XML document and form a tree.
2. Pass the tree as input to a transformation processor.
3. Create a result tree from the XML source tree.
4. Interpret the result tree by comparing the nodes in the tree with the instructions contained in the referenced XSL stylesheet.
5. Format the output.

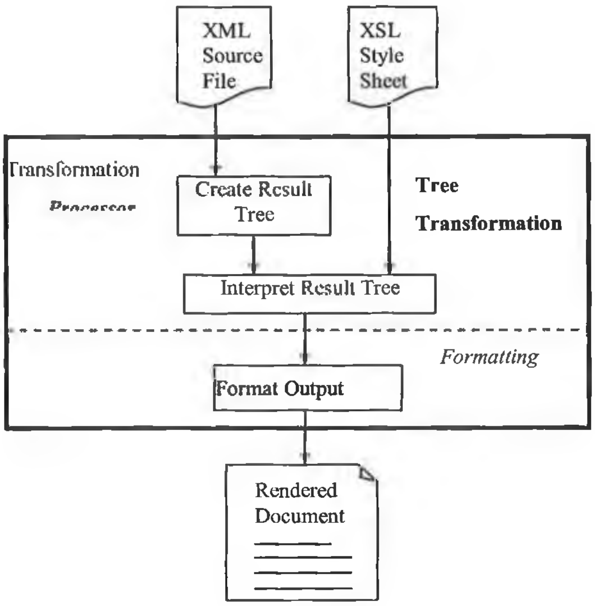
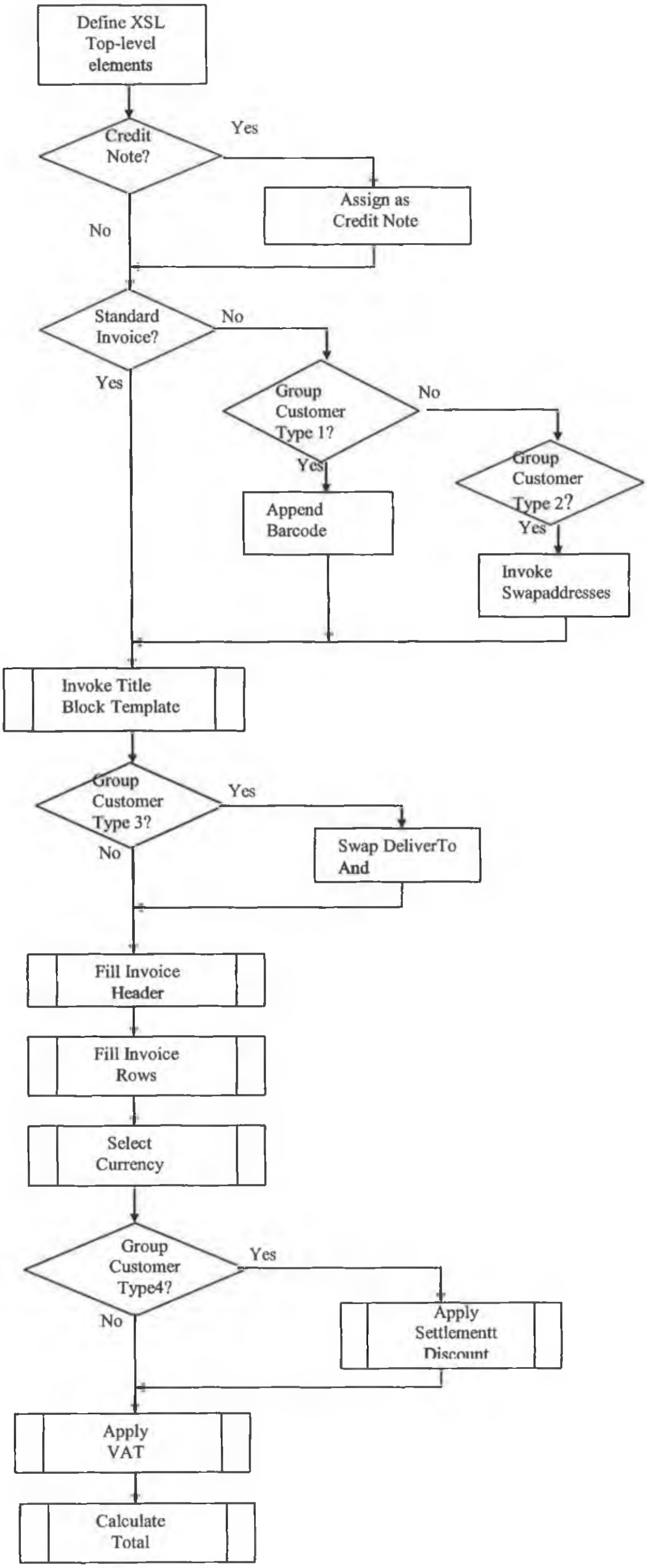


Figure 5.9 XSL Transformation

XSLT documents use templates to describe how to transform an XML document. A template embodies a rule that specifies the node(s) that the template matches and the required output when the pattern is matched. The XSLT processor uses the style sheet to look at each node in the XML document tree. Each node in the XML document is read and the processor compares it with the pattern of each template rule in the style sheet. The processor outputs the template rule when it finds a node that matches the pattern of the template rule.

The flowchart in figure 5.7 outlines the underlying structure of the XSL style sheet developed in order to transform the XML code for invoices/credit notes. The full coding appears in Appendix 4.

Figure 5.10 Flowchart for XSL style sheet



The following extracts outline some of the standard rules that have been implemented in the invoice coding.

<xs:template match="/">

The **<xsl:template>** is an element that defines the rule and the **attribute match="/"** is an XPath expression (see below) that means match at the root node of the XML source tree. The remainder of the code is HTML code that describes the layout.

```
<xsl:template match="/">
                <html>
                        <head>
                                <title>Excel Industries Ltd.
                                        <xsl:value-of
select="/biztalk:biztalk_1/biztalk:body/basda:Invoice/basda:InvoiceHead/basda:InvoiceType" />
                                </title>
                                <style>
                                \:* { behavior: url(#default#VML); }
                .rhdr {color:navy; font-family:arial; font-size:36pt; font-weight:bold; margin-bottom:0px;
padding-bottom:0px;}
                .desc {color:navy; font-family:arial; font-size:9pt; margin-top:0px;}
         {color:black; font-family:arial; font-size:20pt; font-weight:bold;margin-bottom:0px;}
         {color:red; font-family:arial; font-size:20pt; font-weight:bold;}
                .addr {color:black; font-family:arial; font-size:11pt; font-weight:bold;}
                .ftr {color:blck; font-family:arial; font-size:7pt; text-align=center; font-weight:bold; font-
style:italic;margin-bottom:0px; padding-bottom:0px;}
                .bftr {color:navy; font-family:arial; font-size:7pt; text-align=center;margin-bottom:0px;
padding-bottom:0px;margin-top:0px;}
                .hdr {color:black; font-family:arial; font-size:9pt; font-weight:bold;}
                .std {color:black; font-family:arial; font-size:9pt;margin-top:0px;}
                .curr {color:black; font-family:arial; font-size:9pt;}
                .tot {color:black; font-family:arial; font-size:9pt;}
                .btot {color:black; font-family:arial; font-size:9pt; font-weight:bold;}
                .code {color:black; font-family:arial; font-size:9pt; font-weight:bold;}
                                </style>
                        </head>
                        <body bgcolor="#FFFFFF" text="#000000">
                                <table width="100%" border="0">
                                        <tr>
                                                <td width="85%" align="center" valign="top">
                                                        <p class="rhdr">
                                                        INDUSTRIES
                                                        </p>
                                                        <hr style="color:red" width="75%" />
                                                        <p class="desc">
                                                        Industrial Estate, Clonsilla Road, Dublin 15,
Ireland
                                                                <br/>
                                                        :01-8118701 Fax:01-8118777 Our Country Code
is 353. e-mail:sales@excel-industries.com
                                                                <br/>
                                                        Fax Line to Accounts 01-8118786
                                                                <br/>
                                                        Direct 01-8118790 Direct Fax Line 01-8118785
                                                        </p>
                                                </td>
```

<xsl:choose></xsl:choose>

This element selects one of a number of possible outputs based on conditions. Each condition and its associated output is provided by the **xsl:when** child element. The sample text shows the section of code that checks for credit notes.

```
<xsl:variable name="creditnote">
                    <xsl:choose>
                              <xsl:when
test="/biztalk:biztalk_1/biztalk:body/basda:Invoice/basda:InvoiceHead/basda:InvoiceType =
'Sales Invoice'">
                                        <xsl:text>false</xsl:text>
                              </xsl:when>
                              <xsl:otherwise>
                                        <xsl:text>true</xsl:text>
                              </xsl:otherwise>
                    </xsl:choose>
          </xsl:variable>
```

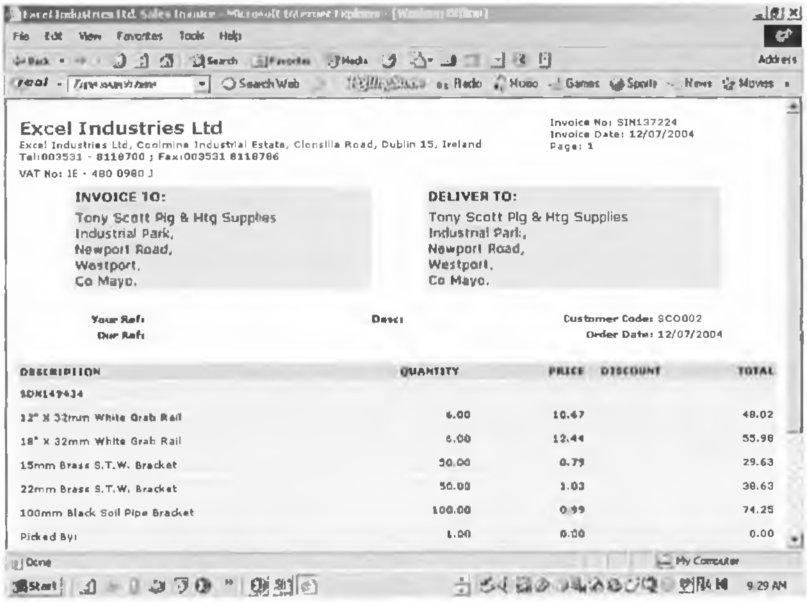
<xsl:if></xsl:if>

The <xsl:if> element changes the output based on a pattern. If the expression is true the contents of the element are output otherwise they are not. The sample code results in an output of “Invoice To” or “Credit To”.

```
            <xsl:if test="$creditnote='false'">
                                                                                    Invoice To:
                                                              </xsl:if>
                                                                              <xsl:if
test="$creditnote='true'">

          Credit To:
```

The figure below shows how the invoice appears in the user agent e.g. browser.



File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Search Web Radio Music Games Sports News Movies
Done
My Computer
Start
9:29 AM

Excel Industries Ltd
Excel Industries Ltd, Coolmine Industrial Estate, Clonsilla Road, Dublin 15, Ireland
Tel:003531 - 8118700 ; Fax:003531 8118786
VAT No: IE - 480 0980 J

Invoice No: SIN137224
Invoice Date: 12/07/2004
Page: 1

INVOICE TO:
Tony Scott Plg & Htg Supplies
Industrial Park,
Newport Road,
Westport,
Co Mayo.

DELIVER TO:
Tony Scott Plg & Htg Supplies
Industrial Park,
Newport Road,
Westport,
Co Mayo.

Your Ref:
Our Ref:
Desc:
Customer Code: SCO002
Order Date: 12/07/2004

DESCRIPTION	QUANTITY	PRICE	DISCOUNT	TOTAL
SON149434				
12" X 32mm White Grab Rail	6.00	10.47		48.02
18" X 32mm White Grab Rail	6.00	12.44		55.98
15mm Brass S.T.W. Bracket	50.00	0.79		29.63
22mm Brass S.T.W. Bracket	50.00	1.03		38.63
100mm Black Soil Pipe Bracket	100.00	0.99		74.25
Picked By:	1.00	0.00		0.00

Figure 5.11Invoice as Rendered by Browser

5.5.2 XPath

The XML Path Language (XPath) is a declarative language defined by the World Wide Web Consortium (W3C) and is used to identify and access portions or subsets of XML documents. XPath uses a path-based syntax similar to that used in file systems and document retrieval. It provides a method that enables the accessing of XML elements, attributes, and other document nodes in a concise and convenient way. XPath operates on the logical structure of an XML document rather than on its syntax. Its primary purpose is to identify sets of nodes that satisfy particular selection criteria.

The key concept to XPath is the *location path* which allows a cascading technique such that each node of the location path is indicated in a directory-like fashion in a manner similar to DOS or UNIX. The node tree replaces the directory tree and each node of the tree has one of the seven *node types* listed above. The basic construct of the location path is a sequence of location steps separated by a slash (/). The example below is taken from the stylesheet developed for the project. The purpose of this *select function* is to locate the node (*element – InvoiceType*) in the XML code.

```
<xsl:value-of
select="/biztalk:biztalk_1/biztalk:body/basda:Invoice/basda:InvoiceHead/basda:InvoiceType
" />
```

This is illustrated by the extract from the tree structure for the XML document shown below.

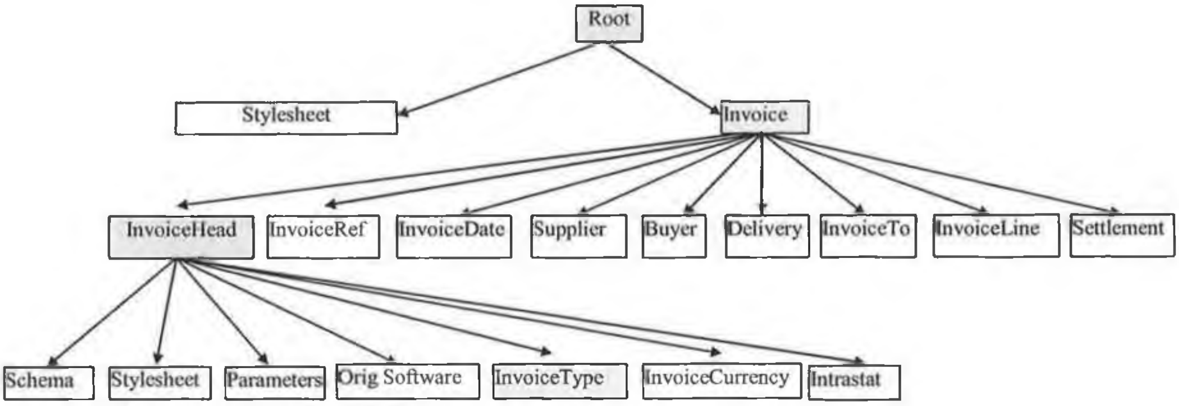


Figure 5.12 Tree Structure of an XML Document

Chapter 6: Implementation

6 Implementation

6.1 Introduction

The implementation of the new system required the adoption of the EAN 128, the configuration of the ADOS application, testing the system under normal working conditions and the roll-out of the updated network infrastructure.

6.2 Overview of New System – Main Business Functions

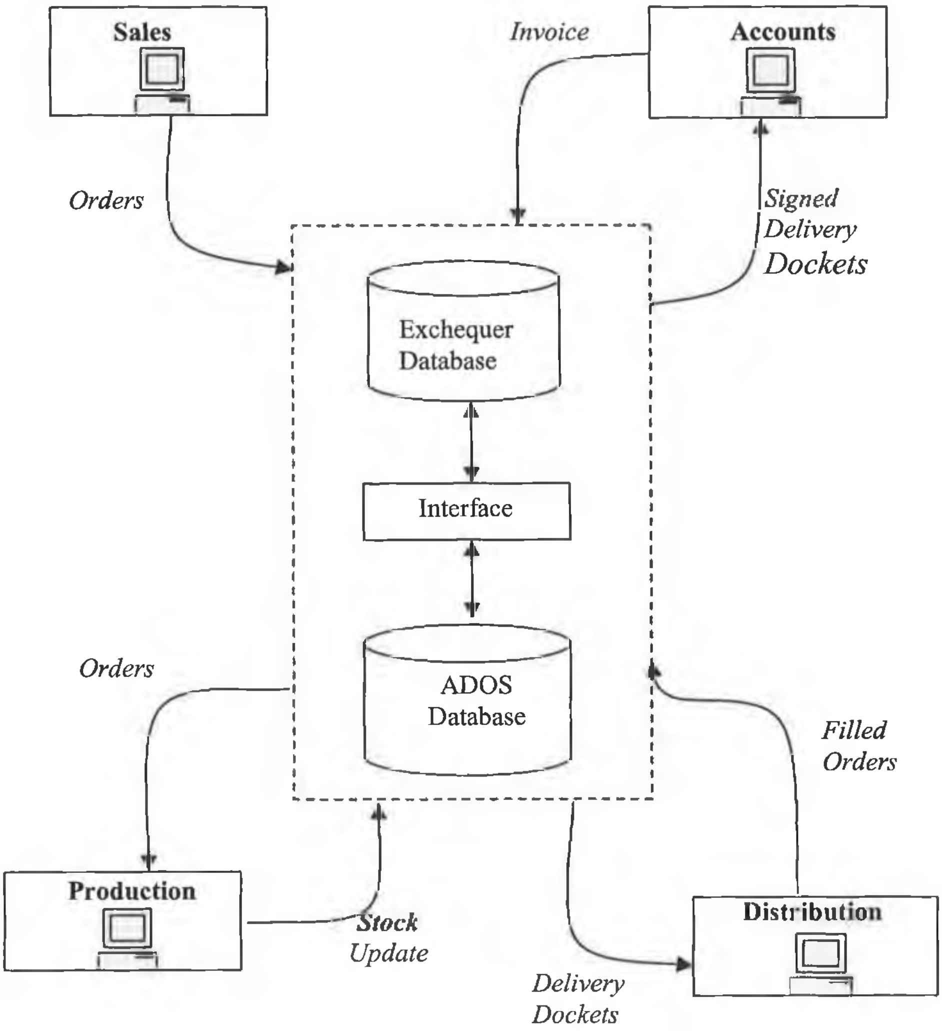


Figure 6.1 Overview of System

The overall strategy was to transform the system to provide a large degree of integration among the various business functions and the Exchequer accounting software application that had been installed in the recent past. The system would be configured such that it is now the repository for all users. Access levels would be set for each function and/or individual. Query handling and ancillary activities would now be processed in soft form. Customers would receive responses via software from fax or e-mail. Once scanned the document is available to everyone in the organization.

6.3 New System Schematic

The schematic diagram shows the manner in which the new document inputting process is integrated into the overall system. The scanned images are temporarily stored in the buffer and then processed in the early hours of the morning. This enables the normal activities to proceed unencumbered by the requirement to process the new data. The buffer permits the delivery docket and invoice details to be stripped out and forwarded for storage.

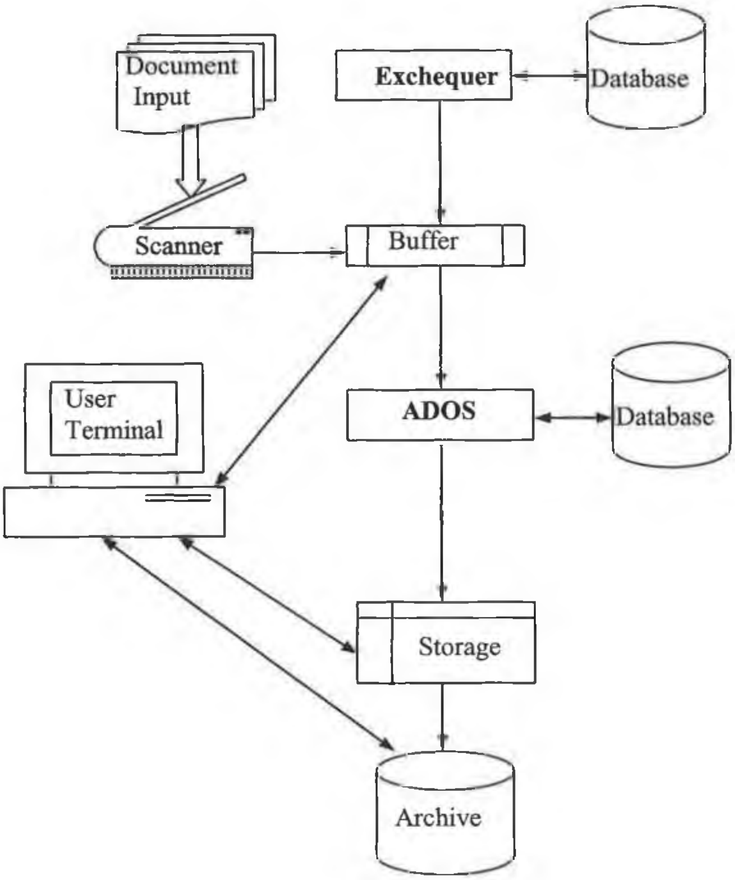


Figure 6.2 Schematic of New System

The form designer facility available with the Exchequer software application was used to design a general delivery docket form set and a range of docket layouts tailored for particular customers based on the scale of business activity and/or business arrangements. This ensures that customer confidentiality would not be compromised as a consequence of the dynamic extraction of data from the system.

This approach underpinned one of the basic elements in the coding i.e. specifying an XSL stylesheet corresponding to a particular customer or group of customers.

6.4 ADOS (Advanced Digital Office System) Overview

The ADOS archive and retrieval system processes, indexes, stores, and manages scanned documents, electronic documents and computer output (COLD).

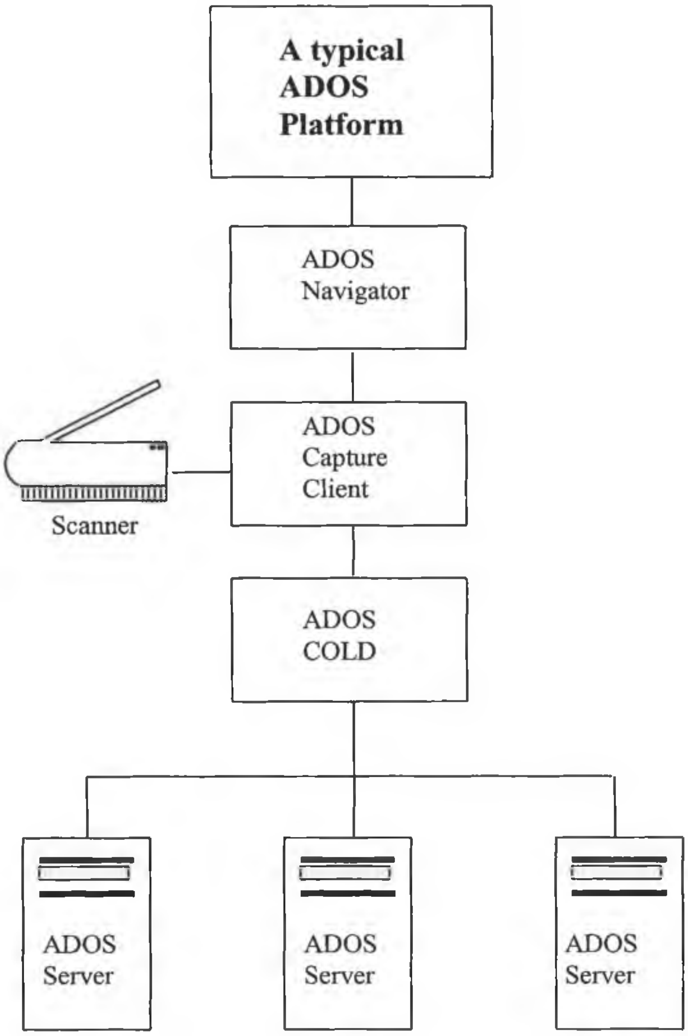


Figure 6.3 Typical ADOS Platform

ADOS was chosen because of its scalability. It offered document security through a centralized back up and version control to maintain the integrity of original documents despite any amendments or changes. Handling data centrally and allowing indexing and archiving at workstations means that large volumes of documents can be managed. The ADOS system stores the archived information in an SQL database which enables the use of database engines e.g. SQL Server.

Scanned and electronic documents are processed and saved locally or remotely.

The modularity of ADOS enabled the system to be tailored to the requirements of the business and so facilitate integration with the existing accounting software.

Features of the system include; scan, process, edit, search, sort, group, view and print among others.

6.5 ADOS Software Modules

6.5.1 ArchiveServer

The ArchiveServer is the information hub in the network. It receives Clients' input, manages the optical storage media and replies to queries from Client Workstations. Communication in the ADOS network takes place via the network protocol NetBIOS. The ADOS server programs e.g. ADOS ArchiveServer and SQLBase server are run under NT. The ADOS ArchiveServer is able communicate with any database that has an SQL interface e.g. DB II, Oracle, etc.

The ArchiveServer carries out the following functions: -

- Registration and labelling of disks and backup storage media.
- Assignment of document categories.
- Configuration: database name and server name, archive buffer, document cache
- Disk naming for optical disks.
- Writing document backups to the assigned optical disks

6.5.2 ArchiveClient

Users log on by means of User Name and Password. Documents are collected in a Multiple Document Interface (MDI) window by scanning either individually or in a batch and saved in the archive buffer.

The System Administrator defines the Document Categories and these categories control the specific manner in which the user enters the Index Data (date, text, integers, amounts and keywords). Controls are built into the entry fields to ensure correct indexing. The controls take the form of Keyword Dictionaries, Indexes or

Default Values. A Document Category Tool Box provides command buttons for the functions of archiving, searching, modifying, deleting, etc.

Use of the MDI allows the documents to be scanned and then distributed to the different document categories (on –screen icons) and then archived.

Retrieving documents involves setting search criteria, which produces a result list, and from this list the relevant documents are chosen. Once loaded from the optical storage media documents can be enlarged, rotated, zoomed and printed if so desired.

The functions of the ADOS ArchiveClient are summarised below.

- Document Capture: -*Scanning, File transfer, Outputting documents available for archiving*
- Archiving Documents: -*Document Category, Document Transfer, Indexing Data, Keyword selection, Archiving, Automatic Bar Code recognition*
- Searching and displaying documents: -*Searching, Selection, MDI Display*
- Processing Documents: - *Document manipulation, File Processing in the original application*

6.5.3 ArchiveAdministrator

System Administrator via the ArchiveAdministrator carries out system configuration. It is also controls user management, access rights and the creation of Document Categories and has the following functions: Creating Document Categories, User Administration and Disk Management

6.5.4 Document Categories

The structure and organisation of the archive is designed around the types of documents, in this instance; delivery dockets, invoices, credit notes and correspondence. Dialog boxes were created for each Document Category using the Document Category Editor.

The Index data fields corresponding to the various Documents and Document Types were entered as Index data in the database. Data types such as date, text, money integer or logical fields were defined. Properties such as height, width, position, assigned keyword dictionary, and mandatory and unique entry were also defined.

6.5.5 User Administration

Access rights have been established for the following levels:

- Database
- Administration
- Access to Document Categories
- Access to individual documents

Users require a Password and User Name to access the database. Administration rights such as "Create Keyword Dictionary" or "Define Document Category" have been granted to specified users to allow them to work with specific functions in ADOS ArchiveAdministrator

Archiving, searching and modifying index data access rights have been allocated to some users to permit them to work with specific Document Categories.

Facilities provided by this module include: -

- User Name and Password assignment
- Access rights based on set criteria
- Group Access Rights

6.5.6 Disk Management

Documents that have been stored in the archive buffer are copied to the Magneto Optical disks for storage and off peak backup. The module controls the registration of the disks and also assigns the Document Categories to the disks for archiving.

6.5.7 ArchiveApi (Application Programming Interface)

This was used to create the customised solution and to enable the integration of the Document Management System into existing system environment.

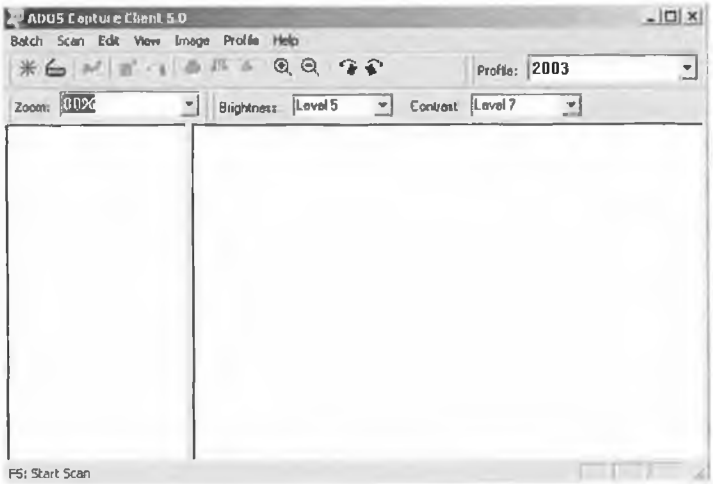
6.5.8 ADOS Implementation- User Interface

The user interface for a typical document input into the ADOS system is shown below.

Once the ADOS application is launched the standard page opens. The settings shown are specifically for the scanning and indexing of Proof of Delivery documentation.

6.6 Graphical User Interface

The delivery dockets are placed in the feeder of a scanner. The user initiates the scanning process (click on icon).

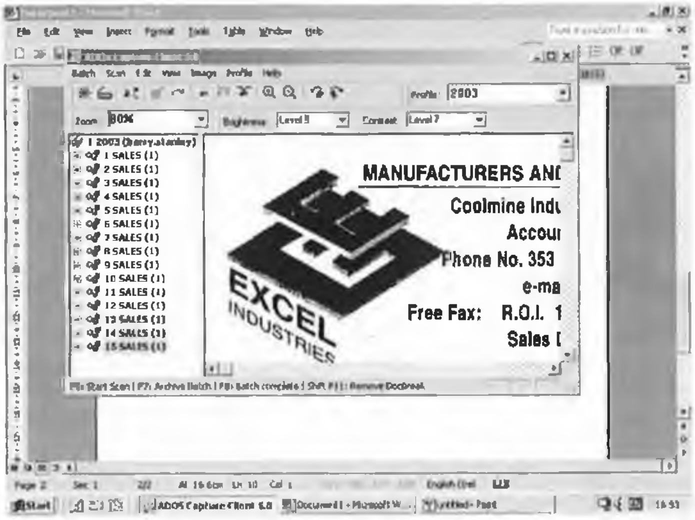


Profile: 2003
Zoom: 80%
Brightness: Level 5
Contrast: Level 7
F5: Start Scan

Figure 6.4 Beginning the Scanning Process

While the scanning is taking place the Kodak software (KOFAX module) reads the barcodes and automatically indexes each document separately.

A green correction mark on the left of each scanned document indicates that the barcode has been read successfully. When all documents are scanned and indexed a green correction mark appears just above the first scanned document next to the profile. This second green mark indicates that documents are now ready to be uploaded.



Profile: 2003
Zoom: 80%
Brightness: Level 5
Contrast: Level 7

1 2003 (barry.stanley)
1 SALES (1)
2 SALES (1)
3 SALES (1)
4 SALES (1)
5 SALES (1)
6 SALES (1)
7 SALES (1)
8 SALES (1)
9 SALES (1)
10 SALES (1)
11 SALES (1)
12 SALES (1)
13 SALES (1)
14 SALES (1)
15 SALES (1)

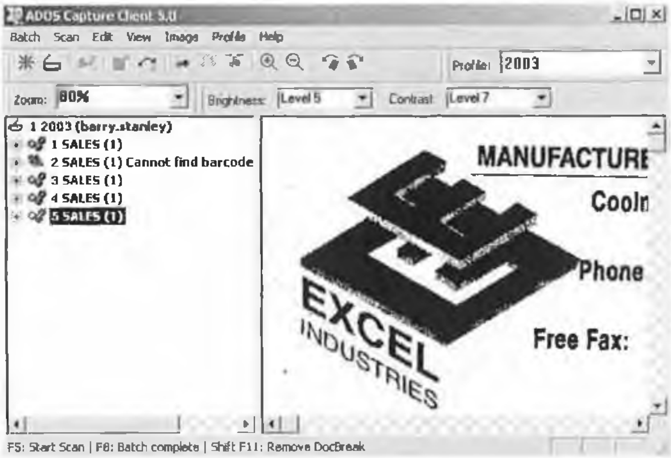
F5: Start Scan | F7: Archive Batch | F8: Batch complete | Shift F11: Remove DocBreak

Page 2 Sec 1 2/2 At 16.6cm Ln 10 Col 1 English (Ire)

Figure 6.5 Documents Successfully Scanned

To upload the scanned and index documents into the ADOS database the user clicks on the icon . This permits instant retrieval by anyone on the network with the necessary access.

In the event that the barcode appended to a given document is not successfully scanned a red arrow appears to highlight this situation.

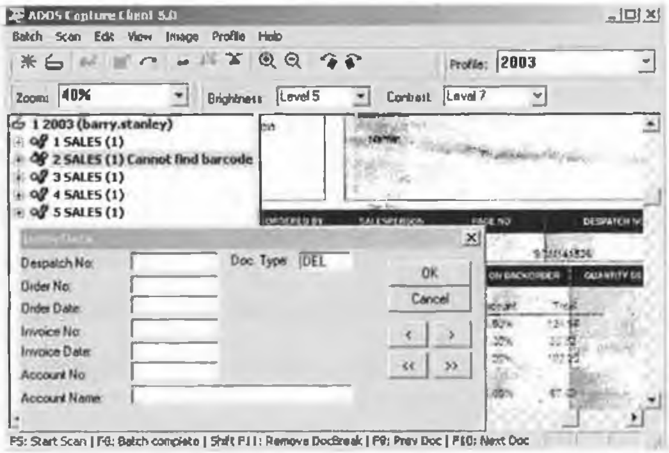


1 2003 (barry.stanley)
1 SALES (1)
2 SALES (1) Cannot find barcode
3 SALES (1)
4 SALES (1)
5 SALES (1)

F5: Start Scan | F8: Batch complete | Shift F11: Remove DocBreak

Figure 6.6 Flagging an Unsuccessful Scan

Clicking on the red arrow enables the user to input the unique despatch number via the manual indexing box shown opposite. The colour of the arrow associated with this document changes to blue to indicate manual input mode.



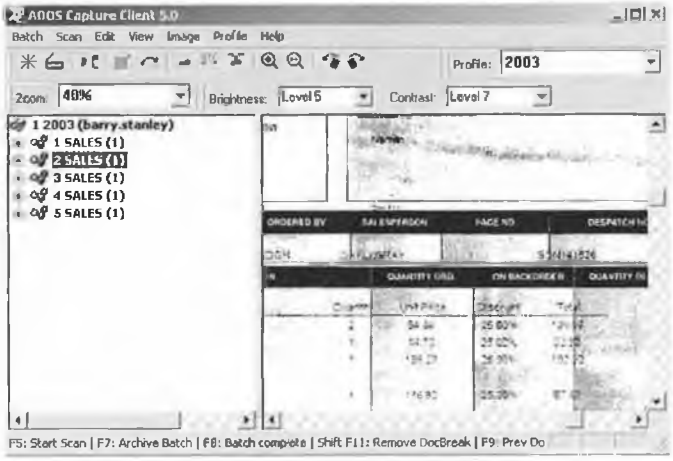
1 2003 (barry.stanley)
1 SALES (1)
2 SALES (1) Cannot find barcode
3 SALES (1)
4 SALES (1)
5 SALES (1)

Despatch No:
Order No:
Order Date:
Invoice No:
Invoice Date:
Account No:
Account Name:
Doc Type: DEL

F5: Start Scan | F8: Batch complete | Shift F11: Remove DocBreak | F9: Prev Doc | F10: Next Doc

Figure 6.7 Inputting the despatch number

Once the despatch number is inputted and confirmed (click okay) the blue arrow changes to a green correction mark. A second correction mark now appears by the profile line to indicate that all documents can be uploaded.

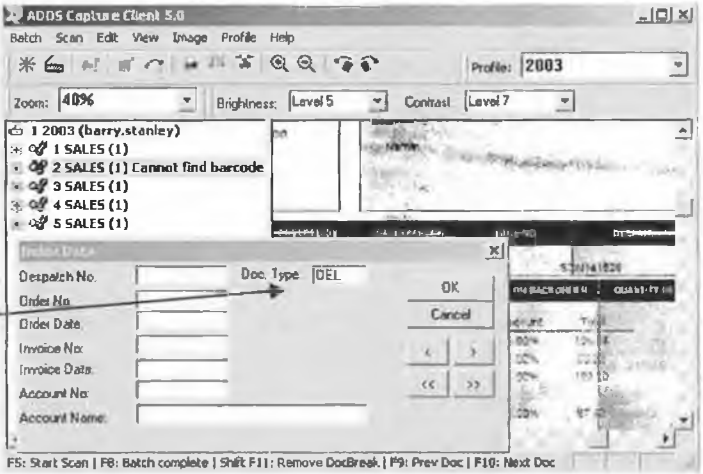


1 2003 (barry.stanley)
1 SALES (1)
2 SALES (1)
3 SALES (1)
4 SALES (1)
5 SALES (1)

F5: Start Scan | F7: Archive Batch | F8: Batch complete | Shift F11: Remove DocBreak | F9: Prev Do

Figure 6.8 Uploading the Documents

Collection dockets are handled in a similar way to Proof of Delivery dockets. To do this the index fields needed to be completed and the "DEL" replaced by "COL"



Profile: 2003
Zoom: 40%
Brightness: Level 5
Contrast: Level 7

1 2003 (barry.stanley)
1 SALES (1)
2 SALES (1) Cannot find barcode
3 SALES (1)
4 SALES (1)
5 SALES (1)

Despatch No.
Order No.
Order Date
Invoice No.
Invoice Date
Account No.
Account Name
Doc. Type DEL

F5: Start Scan | F8: Batch complete | Shift F11: Remove DocBreak | F9: Prev Doc | F10: Next Doc

Figure 6.9 Handling Collection Dockets

6.7 Network Infrastructure Upgrade

The improvements in the network infrastructure were implemented in parallel with the development of the new DMS. The previous and new configurations are shown in figures 6.10 and 6.11 on the following pages. The significant alterations were carried out in anticipation of larger data traffic and the concomitant the need for greater bandwidth and the desire to future-proof the network. The core of the system comprises four servers running a combination of software that ranges from Microsoft SBS to the ADOS document managing system. The principal node on the network is a cluster of three 3300MM, 3com switches.

A variety of user hardware is supported on the network from PCs to printers and other peripheral devices. The table below details the number and location of the devices.

	Hardware			
Location	PC	Printer	Scanner	Plotter
General Manager	1	1		
Sales	4	1		
Marketing	2	1		
Production	4			
Distribution	6	1		
Accounts	4	1		
Research &Development	6	1	1	1

Table6.1 Devices on the Network

Heavy emphasis has been placed on the need for back-up and recovery. All data on the ADOS system are written to an optical drive and than backed up to tape (Brightstore 130GB). Each server is shadowed and an Incremental back up is carried out every day on all server drives and shadows. Full back up is undertaken once each week. A remote server has been installed off-site which shadows and backs up all the other servers.

Servers	Back up
3 x Dell Power Edge (1 off-site)	AIT SDX 500C
	Sony HP 9100 (Optical)

Table 6.2 Network Servers and Backup

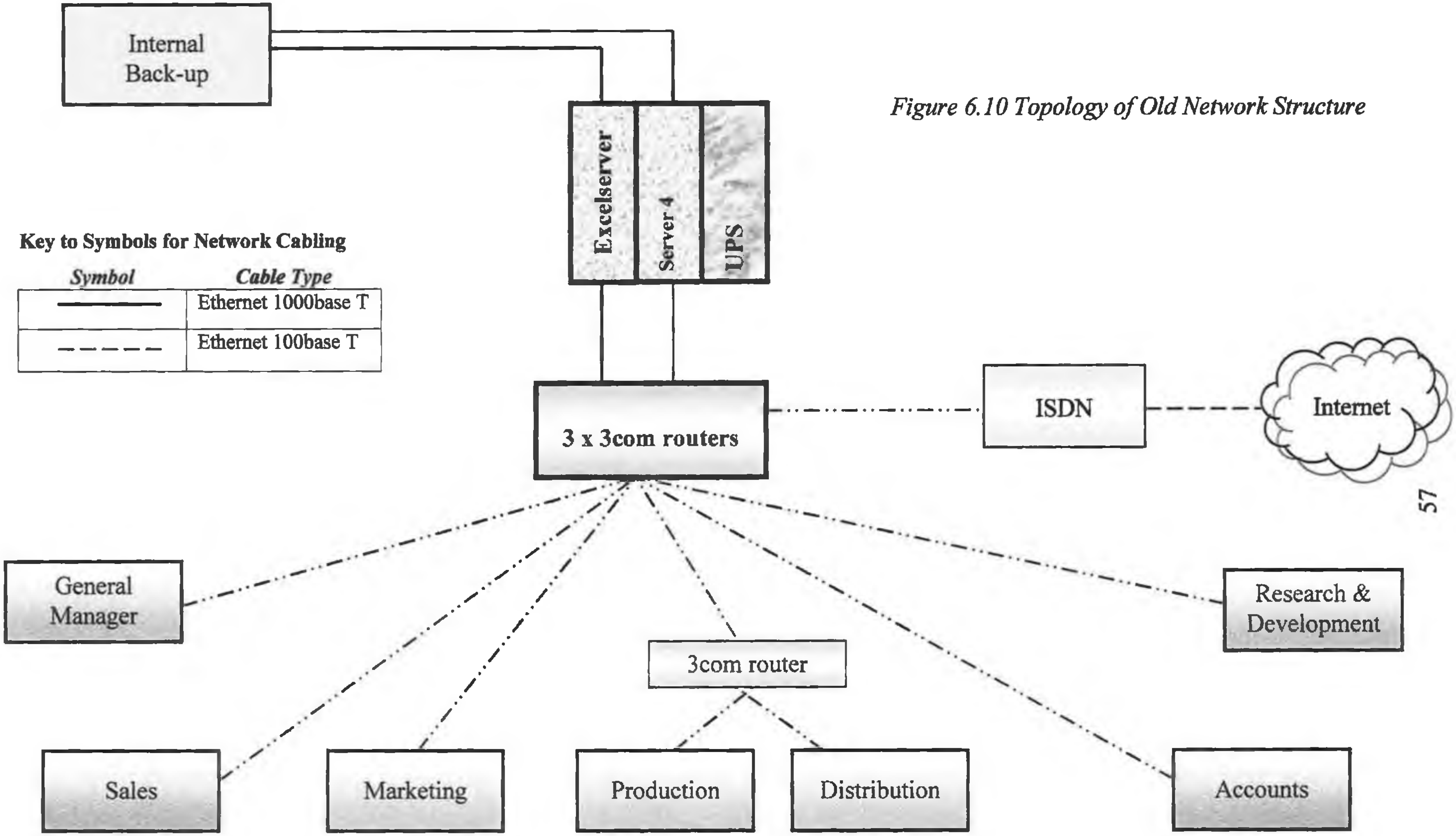
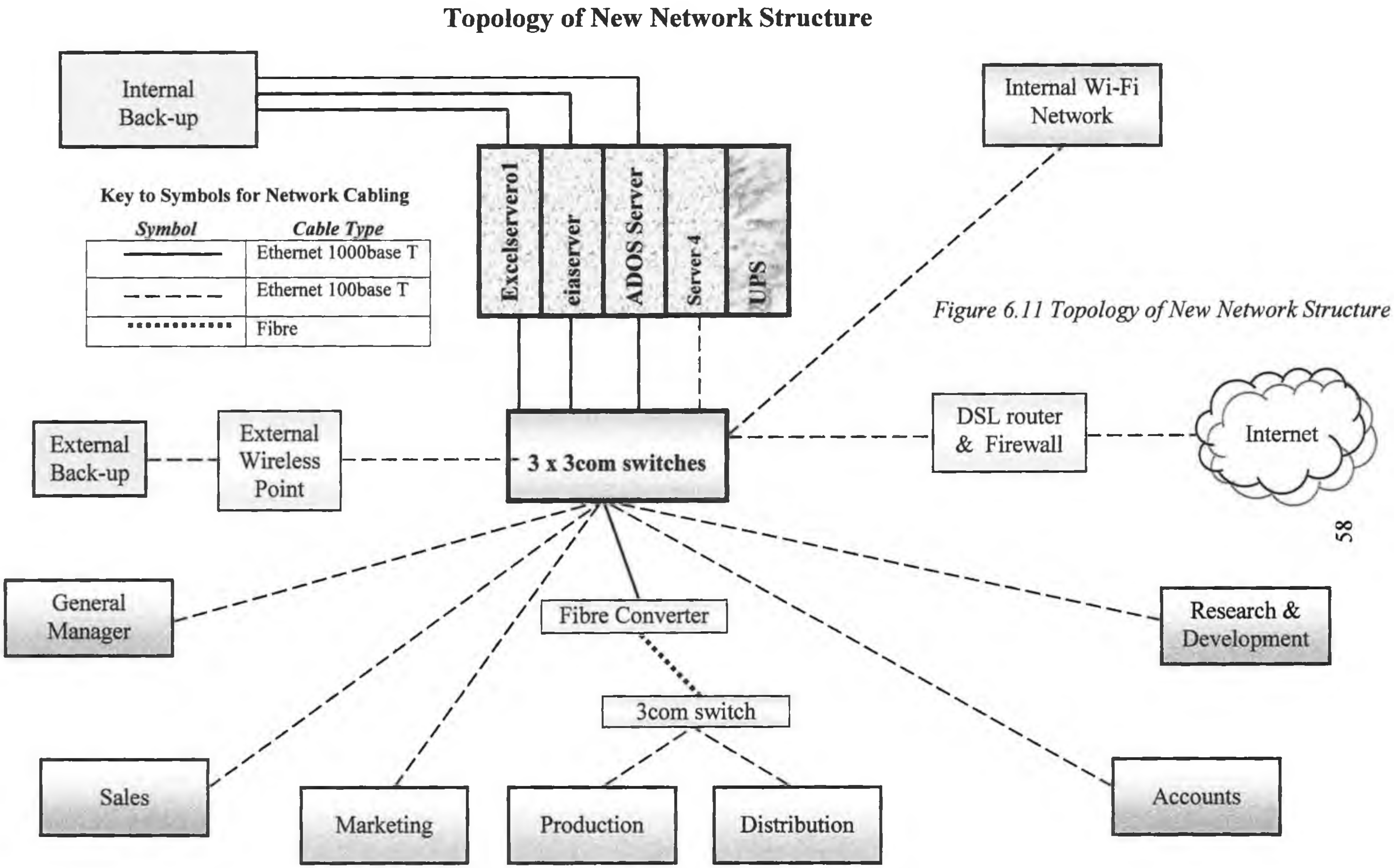


Figure 6.10 Topology of Old Network Structure



Key to Symbols for Network Cabling

Symbol	*Cable Type*
———	Ethernet 1000base T
– – – –	Ethernet 100base T
···········	Fibre

Figure 6.11 Topology of New Network Structure

Chapter 7: Conclusion

7 Conclusion

7.1 Project Review

The original objective of the project was "*the implementation of a Document Management System (DMS) to allow Excel Industries improve their response to customer account queries. The system would allow them to collect payments faster and reduce their Debtors' Days Outstanding (DDO) figures and improve productivity.*"

This general objective was expanded and clarified as detailed in chapter 4. Thus the aims of the project evolved into the following:

- All new and historic delivery notes to be scanned into the system and indexed.
- Extraction of Sales Invoices and Credit Notes from the Exchequer software during a daily run and the importation of these into the DMS.
- All supplier correspondence to be stored in the DMS
- All data extracted from the Exchequer accounting package to be available from the DMS.
- Upgrading of the network to facilitate the proposed system.

The limited objective at the outset for the project turned out to be merely a staging post in the overall process. Fortunately, the considerable patience exhibited on the part of the Excel Industries management allowed the project "to breathe". In spite of the poorly defined requirements the evolutionary model incorporating prototyping proved to be invaluable in the development of the system. There can be no denying that the integration of the new system and the existing accounting software would never have been suggested had this evolutionary process not been undertaken. This proposal represented a significant shift from the original brief for the project.

On the whole, the project required a dynamic and flexible development and this is demonstrated in the evolution and maturing of the specification, which reflects the distillation of ideas that occurred as myself and the company moved up the knowledge curve and realised the potential of the development.

The decision to opt for the BASDA schema helped to formalise the approach to the code development and opens up the possibilities for other opportunities in the area of web-based commerce knowledge curve and realised the potential of the development?

Excel Industries have made considerable reductions in response times to customer queries. One surprising and ironic outcome of the implementation of the new system was a reduction in request for documentary proof. The more speedily a customer could be furnished with the proof s/he required the less likely that it would be sought in the first place resulting in a reduction of demand on the system and faster throughput of payment leading to a reduction in debtors' days and improved cash flow. In addition, electronic formats have led to a decrease in the amount of physical space required for file storage and the elimination of the tedious requirement for manual filing is greatly appreciated by the staff.

The network upgrade has resulted in a system that has greater capacity and is more robust in structure. The internal Wi-fi network has enabled the system to have greater flexibility. The capacity of on-site back up has been expanded considerably. A heavier emphasis has been placed in relation to off-site back up and storage. What otherwise might have been a tight program to achieve was to some extent provided with a longer lead-in because of the evolutionary development of the required software.

7.2 The Future

As alluded to in the previous section greater openings now exist for web-based commerce. Standard schema permit more formalized exchange of data. The XML coding significantly reduces file sizes thus permitting faster data transfer. This is of particular importance in the area of business-to-business (b2b) commerce. It is hoped to lever added value from the effort thus expended by developing new features and enhancing existing features particularly in the sphere of web-based commercial activities. This will require a combination of technical and commercial developments. The former will necessitate a major overhaul and expansion of the existing website. This is presently under investigation. The outcome should enable the following features:

- On-line catalogue
- On-line ordering by sales staff.
- On-line ordering by trusted partners/customers.

Particular attention will have to be paid to security if this last item is to come to fruition. In addition, it will be necessary to identify suitable partners to undertake the trialling of such a development.

One other feature under consideration is the integration of the accounting system with an on-line catalogue/ordering process. It is envisaged that it will be possible to generate invoices based on a twenty-four hour accounts up-date. This in turn could be interfaced with stock control to enable customers to order on the basis of the most up-to-date stock levels, and permit the presentation to reflect the terms on which they do business, i.e. discount offered, terms of settlement, etc.

Appendix 1

Bar Code Verification

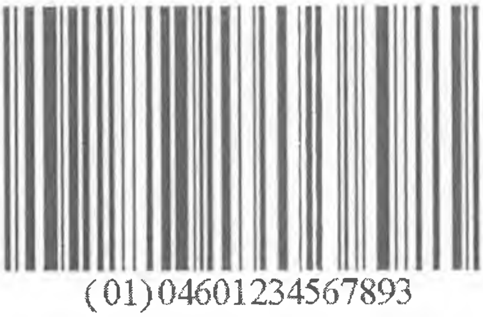


Figure 1A EAN-128 Symbol

It is the responsibility of the 'originator' of the bar code symbol to ensure that it meets the quality requirements of the entire supply chain.[EAN Bar Code Verification for Linear Symbologies, 2003]

- Verification: *The technical process by which a bar code symbol is measured to determine its conformance with the specification for that symbol.*

The main job of verification is to check that the quality grade meets minimum requirements. Verification of an UCC/EAN-128 symbol requires checking that the symbol structure is correct, that it is dimensionally consistent with the EAN/UCC requirements and that the data is correctly formatted. This includes ensuring the correct use of the FNC1 symbol character. Verification of any symbol involves ensuring that the amounts of light of the specified colour reflected by the bars and spaces respectively are sufficiently distinct to enable the symbol to be correctly recognised by a scanning system. 100% sampling is not necessary. Sample verification is the norm.

EAN 128 decoding relies primarily on edge-to-similar edge distances and so it is relatively immune to large amounts of consistent gain or loss across the symbol. The edge-to-similar edge distances are measured from the leading edge of one bar to the leading edge of the next (or from one trailing edge to the next) and these tend to move in the same direction if there is bar width gain or loss.

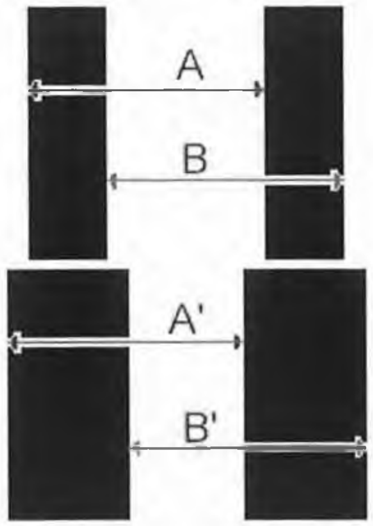


Figure 1.B   Edge to similar edge measurements

Hand-scanned verifiers

The bar code reader comprises both scanner and decoder components. A bar code scanner collects light reflected from the bars and spaces of the symbol and outputs an electric signal proportional to the amount of light reflected from each and every bar and space. The decoder interprets the electrical signals by applying the decode algorithm to decide what data has been encoded.

The scan heads containing the optical components can be of different types from device to device but the operating principle is the same. The scan head must be moved manually across the symbol to generate the scanning action.

Obtaining a Correct Scan

The symbol must be lying on a flat surface in order to achieve a smooth scan. Bumps or irregularities can lead to unpredictable and inaccurate results. Pass the scanner smoothly at a reasonable speed across the symbol up to ten times, each time traversing a different part of the symbol.

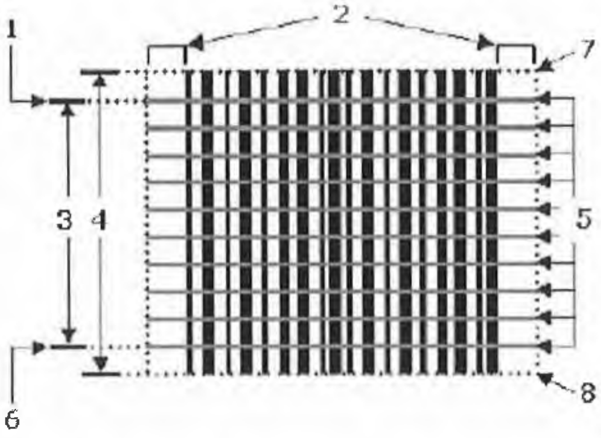


Figure 1.C Areas to Scan for Verification

The scan path exits the top or bottom of the symbol resulting in mis-scans, or short reads.

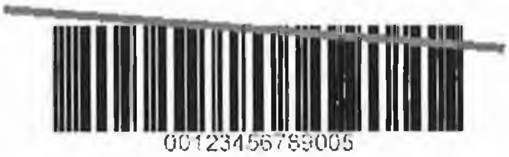


Figure 1.D

The scan path runs too close to the top or bottom edge of the symbol resulting in poor modulation values due to interference from the light area above or below the symbol.

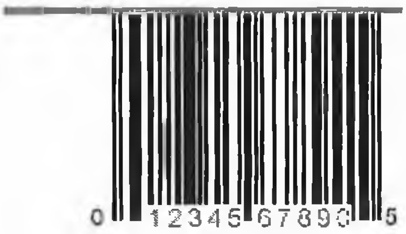


Figure 1.E

Irregular or curved scanning motion results in acceleration or deceleration during the scan and leads to varying Decodability values.

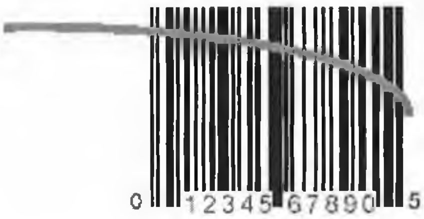


Figure 1.F

The scan path starts or finishes too close to the symbol leading to failure to decode quiet zone and low Decodability grading.

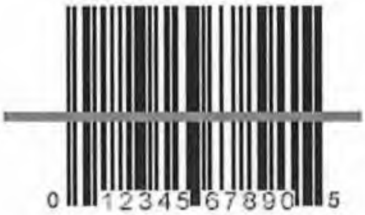


Figure 1.G

Appendix 2

BASDA Schema

```
<!--  From http://www.basda.org/schema/eBIS-XMLschema_invoice_v3.01.xml -->
<Schema name="eBIS-XML_Invoice_v3.01.xml" xmlns="urn:schemas-microsoft-
com:xml-data
" xmlns:dt="urn:schemas-microsoft-com:datatypes">
<ElementType name="Invoice" content="eltOnly" order="seq">
            <element type="InvoiceHead"/>
            <element type="InvoiceReferences"/>
            <element type="InvoiceDate"/>
            <element type="TypeOfSupply" minOccurs="0" maxOccurs="1"/>
            <element type="TaxPointDate" minOccurs="0" maxOccurs="1"/>
            <element type="Supplier"/>
            <element type="Buyer        " minOccurs="0" maxOccurs="1"/>
            <element type="Delivery" minOccurs="0" maxOccurs="*"/>
            <element type="InvoiceTo"/>
            <element type="InvoiceLine" minOccurs="1" maxOccurs="*"/>
            <element type="PercentDiscount" minOccurs="0" maxOccurs="*"/>
            <element type="AmountDiscount        " minOccurs="0" maxOccurs="*"/>
            <element type="SpecialInstructions" minOccurs="0" maxOccurs="*"/>
            <element type="Narrative" minOccurs="0" maxOccurs="*"/>
            <element type="Settlement" minOccurs="0" maxOccurs="1"/>
            <element type="TaxSubTotal" minOccurs="1" maxOccurs="*"/>
            <element type="InvoiceTotal"/>
</ElementType>
<ElementType name="InvoiceDate" content="textOnly" dt:type="dateTime"/>
<ElementType name="InvoiceReferences" content="eltOnly" order="seq">
            <element type="ContractOrderReference" minOccurs="0" maxOccurs="1"/>
            <element type="CostCentre" minOccurs="0" maxOccurs="1"/>
            <element type="BuyersOrderNumber" minOccurs="0" maxOccurs="1"/>
            <element type="Department" minOccurs="0" maxOccurs="1"/>
            <element type="GeneralLedgerCode" minOccurs="0" maxOccurs="1"/>
            <element type="ProjectCode" minOccurs="0" maxOccurs="1"/>
            <element type="ProjectAnalysisCode" minOccurs="0" maxOccurs="1"/>
            <element type="SuppliersInvoiceNumber"/>
</ElementType>
<ElementType name="ContractOrderReference" content="textOnly" dt:type="string"/>
<ElementType name="CostCentre" content="textOnly" dt:type="string"/>
<ElementType name="Department" content="textOnly" dt:type="string"/>
<ElementType name="GeneralLedgerCode" content="textOnly" dt:type="string"/>
<ElementType name="ProjectCode" content="textOnly" dt:type="string">
            <AttributeType name="Preserve" dt:type="enumeration" dt:values="true"
default="true"/>
                        <attribute type="Preserve"/>
</ElementType>
<ElementType name="ProjectAnalysisCode" content="textOnly" dt:type="string">
            <AttributeType name="Preserve" dt:type="enumeration" dt:values="true"
default="true"/>
                        <attribute type="Preserve"/>
</ElementType>
<ElementType name="InvoiceHead" content="eltOnly" order="seq">
            <element type="Schema"/>
            <element type="Stylesheet" minOccurs="0" maxOccurs="1"/>
            <element type="Parameters"/>
            <element type="OriginatingSoftware" minOccurs="0" maxOccurs="1"/>
            <element type="TestFlag" minOccurs="0" maxOccurs="1"/>
            <element type="InvoiceType"/>
```

```
          <element type="Function" minOccurs="0" maxOccurs="1"/>
          <element type="InvoiceCurrency"/>
          <element type="Intrastat" minOccurs="0" maxOccurs="1"/>
          <element type="Checksum"/>
</ElementType>
<ElementType name="InvoiceType" content="textOnly" dt:type="string">
          <AttributeType name="Code" dt:type="string" default="PUO"/>
          <AttributeType name="Codelist" dt:type="string" default="BASDA"/>
                    attribute type="Code"/>
                    attribute type="Codelist"/>
</ElementType>
<ElementType name="Function" content="textOnly" dt:type="string">
          <AttributeType name="Code dt:type="string default="FIO"/>
          <AttributeType name="Codelist" dt:type="string" default="BASDA"/>
                    <attribute type="Code"/>
                    <attribute type="Codelist"/>
</ElementType>
<ElementType name="Supplier" content="eltOnly" order="seq">
          <element type="SupplierReferences" minOccurs="0" maxOccurs="1"/>
          <element type="Party" minOccurs="1" maxOccurs="*"/>
          <element type="Address" minOccurs="0" maxOccurs="1"/>
          <element type="Contact" minOccurs="0" maxOccurs="*"/>
</ElementType>
<ElementType name="Buyer" content="eltOnly" order="seq">
          <element type="BuyerReferences" minOccurs="0" maxOccurs="1"/>
          <element type="Party" minOccurs="1" maxOccurs="*"/>
          <element type="Address" minOccurs="0" maxOccurs="1"/>
          <element type="Contact" minOccurs="0" maxOccurs="*"/>
</ElementType>
<ElementType name="DeliverTo" content="eltOnly" order="seq">
          <element type="DeliverToReferences" minOccurs="0" maxOccurs="1"/>
          <element type="Party" minOccurs="1" maxOccurs="*"/>
          <element type="Address" minOccurs="0" maxOccurs="1"/>
          <element type="Location" minOccurs="0" maxOccurs="1"/>
          <element type="Contact" minOccurs="0" maxOccurs="*"/>
</ElementType>
<ElementType name="InvoiceTo" content="eltOnly" order="seq">
          <element type="InvoiceToReferences" minOccurs="0" maxOccurs="1"/>
          <element type="Party" minOccurs="1" maxOccurs="*"/>
          <element type="Address" minOccurs="0" maxOccurs="1"/>
          <element type="Contact" minOccurs="0" maxOccurs="*"/>
</ElementType>
<ElementType name="Carrier" content="eltOnly" order="seq">
          <element type="CarrierReferences" minOccurs="0" maxOccurs="1"/>
          <element type="Party" minOccurs="1" maxOccurs="*"/>
          <element type="Address" minOccurs="0"/>
          <element type="Contact" minOccurs="0" maxOccurs="*"/>
</ElementType>
<ElementType name="Delivery" content="eltOnly" order="seq">
          <element type="DeliverTo" minOccurs="0" maxOccurs="1"/>
          <element type="Carrier" minOccurs="0" maxOccurs="1"/>
          <element type="Quantity" minOccurs="0" maxOccurs="1"/>
          <element type="EarliestAcceptableDate" minOccurs="0" maxOccurs="1"/>
          <element type="LatestAcceptableDate" minOccurs="0" maxOccurs="1"/>
          <element type="PreferredDate" minOccurs="0" maxOccurs="1"/>
          <element type="SpecialInstructions" minOccurs="0" maxOccurs="*"/>
          <element type="DeliveryInformation" minOccurs="0" maxOccurs="*"/>
          <element type="Narrative" minOccurs="0" maxOccurs="*"/>
</ElementType>
<ElementType name="EarliestAcceptableDate" content="textOnly" dt:type="dateTime"/>
<ElementType name="LatestAcceptableDate" content="textOnly" dt:type="dateTime"/>
<ElementType name="PreferredDate" content="textOnly" dt:type="dateTime"/>
<ElementType name="InvoiceLine" content="eltOnly" order="seq">
          <AttributeType name="Action" dt:type="enumeration" dt:values="Add Delete
Change" default="Add"/>
          <AttributeType name="TypeCode" dt:type="string"/>
          <AttributeType name="TypeDescription" dt:type="string"/>
          <AttributeType name="TypeCodelist" dt:type="string"/>
          <attribute type="Action"/>
          <attribute type="TypeCode"/>
          <attribute type="TypeDescription"/>
          <attribute type="TypeCodelist"/>
          <element type="LineNumber"/>
          <element type="InvoiceLineReferences" minOccurs="0" maxOccurs="1"/>
          <element type="Product" minOccurs="0" maxOccurs="*"/>
          <element type="Quantity" minOccurs="0" maxOccurs="1"/>
          <element type="Price" minOccurs="1" maxOccurs="*"/>
          <element type="PercentDiscount" minOccurs="0" maxOccurs="*"/>
          <element type="AmountDiscount" minOccurs="0" maxOccurs="*"/>
          <element type="LineTax" minOccurs="0" maxOccurs="1"/>
          <element type="LineTotal"/>
          <element type="Delivery" minOccurs="0" maxOccurs="*"/>
          <element type="InvoiceLineInformation" minOccurs="0" maxOccurs="*"/>
          <element type="ExtendedDescription" minOccurs="0" maxOccurs="*"/>
          <element type="Narrative" minOccurs="0" maxOccurs="*"/>
          <element type="LotSerial" minOccurs="0" maxOccurs="*"/>
          <element type="Intrastat" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="LineNumber" content="textOnly" dt:type="int">
          AttributeType name="Preserve" dt:type="enumeration" dt:values="true"
default="true"/>
                    <attribute type="Preserve"/>
</ElementType>
<ElementType name="LineTotal" content="textOnly" dt:type="float"/>
<ElementType name="InvoiceLineReferences" content="eltOnly" order="seq">
          <lement type="ContractOrderReference" minOccurs="0" maxOccurs="1"/>
          <element type="CostCentre" minOccurs="0" maxOccurs="1"/>
          <element type="GeneralLedgerCode" minOccurs="0" maxOccurs="1"/>
          <element type="ProjectCode" minOccurs="0" maxOccurs="1"/>
          <element type="ProjectAnalysisCode" minOccurs="0" maxOccurs="1"/>
          <element type="BuyersOrderNumber" minOccurs="0" maxOccurs="1"/>
          <element type="OrderLineNumber"/>
          <element type="BuyersOrderLineReference" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="PercentDiscount" content="eltOnly" order="seq">
          <lement type="QualifyingTerms" minOccurs="0" maxOccurs="1"/>
          <element type="Type" minOccurs="0" maxOccurs="1"/>
          <element type="Percentage"/>
</ElementType>
<ElementType name="AmountDiscount" content="eltOnly" order="seq">
          <element type="QualifyingTerms" minOccurs="0" maxOccurs="1"/>
          <element type="Type" minOccurs="0" maxOccurs="1"/>
          <element type="Amount"/>
</ElementType>
```

```
<ElementType name="Narrative" content="textOnly"/>
<ElementType name="Product" content="eltOnly" order="seq">
        <element type="SuppliersProductCode" minOccurs="0" maxOccurs="1"/>
        <element type="BuyersProductCode" minOccurs="0" maxOccurs="1"/>
        <element type="TradedUnitCode" minOccurs="0" maxOccurs="1"/>
        <element type="ConsumerUnitCode" minOccurs="0" maxOccurs="1"/>
        <element type="Description" minOccurs="0" maxOccurs="1"/>
        <element type="Properties" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="SuppliersProductCode" content="textOnly" dt:type="string"/>
<ElementType name="Description" content="textOnly" dt:type="string"/>
<ElementType name="Quantity" content="eltOnly" order="seq">
        <AttributeType name="UOMCode" dt:type="string"/>
        <AttributeType name="UOMDescription" dt:type="string"/>
        <AttributeType name="UOMCodelist" dt:type="string" default="UNECE"/>
                <attribute type="UOMCode"/>
                <attribute type="UOMDescription"/>
                <attribute type="UOMCodelist"/>
                <element type="Packsize" minOccurs="0" maxOccurs="1"/>
                <element type="Amount" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="Price" content="eltOnly" order="seq">
        <AttributeType name="UOMCode" dt:type="string"/>
        <AttributeType name="UOMDescription" dt:type="string"/>
        <AttributeType name="UOMCodelist" dt:type="string" default="UNECE"/>
        <attribute type="UOMCode"/>
        <attribute type="UOMDescription"/>
        <attribute type="UOMCodelist"/>
        <element type="Units" minOccurs="0" maxOccurs="1"/>
        <element type="UnitPrice"/>
        <element type="SpecialPriceDescription" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="Address" content="eltOnly" order="seq">
        element type="AddressLine" minOccurs="0" maxOccurs="*"/>
        <element type="Street" minOccurs="0" maxOccurs="*"/>
        <element type="City" minOccurs="0" maxOccurs="1"/>
        <element type="State" minOccurs="0" maxOccurs="1"/>
        <element type="PostCode" minOccurs="0" maxOccurs="1"/>
        <element type="Country" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="Party" content="textOnly" dt:type="string"/>
<ElementType name="Street" content="textOnly" dt:type="string"/>
<ElementType name="City" content="textOnly" dt:type="string"/>
<ElementType name="State" content="textOnly" dt:type="string"/>
<ElementType name="PostCode" content="textOnly" dt:type="string"/>
<ElementType name="Country" content="textOnly" dt:type="string">
        <AttributeType name="Code" dt:type="string"/>
        <AttributeType name="Codelist" dt:type="string" default="ISO3166-1/3"/>
                <attribute type="Code"/>
                <attribute type="Codelist"/>
</ElementType>
<ElementType name="AddressLine" content="textOnly" dt:type="string"/>
<ElementType name="Contact" content="eltOnly" order="seq">
        <element type="Name" minOccurs="0" maxOccurs="1"/>
        <element type="Department" minOccurs="0" maxOccurs="1"/>
        <element type="InternalAddress" minOccurs="0" maxOccurs="1"/>
        <element type="DDI" minOccurs="0" maxOccurs="1"/>
        <element type="Switchboard" minOccurs="0" maxOccurs="1"/>
        <element type="Fax" minOccurs="0" maxOccurs="1"/>
        <element type="Telex" minOccurs="0" maxOccurs="1"/>
        <element type="Email" minOccurs="0" maxOccurs="1"/>
        <element type="Mobile" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="Name" content="textOnly" dt:type="string"/>
<ElementType name="InternalAddress" content="textOnly" dt:type="string"/>
<ElementType name="LineTax" content="eltOnly" order="seq">
        <element type="MixedRateIndicator" minOccurs="0" maxOccurs="1"/>
        <element type="TaxRate"/>
        <element type="TaxValue" minOccurs="0" maxOccurs="1"/>
        <element type="TaxRef" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="MixedRateIndicator" content="textOnly" dt:type="string"/>
<ElementType name="TaxRate" content="textOnly" dt:type="float">
        <AttributeType name="Code" dt:type="string" required="yes"/>
        <AttributeType name="Codelist" dt:type="string" default="BASDA"/>
                <attribute type="Code"/>
                <attribute type="Codelist"/>
</ElementType>
<ElementType name="TaxValue" content="textOnly" dt:type="float"/>
<ElementType name="TaxRef" content="textOnly" dt:type="string">
        <AttributeType name="Code" dt:type="string"/>
        <AttributeType name="Codelist" dt:type="string" default="BASDA"/>
                <attribute type="Code"/>
                <attribute type="Codelist"/>
</ElementType>
<ElementType name="InvoiceTotal" content="eltOnly" order="seq">
        <element type="NumberOfLines"/>
        <element type="NumberOfTaxRates"/>
        <element type="LineValueTotal"/>
        <element type="FreightCharges" minOccurs="0" maxOccurs="1"/>
        <element type="MiscCharges" minOccurs="0" maxOccurs="1"/>
        <element type="SettlementDiscountTotal" minOccurs="0" maxOccurs="1"/>
        <element type="TaxableTotal"/>
        <element type="TaxTotal"/>
        <element type="NetPaymentTotal"/>
        <element type="GrossPaymentTotal"/>
</ElementType>
<ElementType name="Checksum" content="textOnly" dt:type="float"/>
<ElementType name="FreightCharges" content="textOnly" dt:type="float"/>
<ElementType name="MiscCharges" content="textOnly" dt:type="float"/>
<ElementType name="TaxTotal" content="textOnly" dt:type="float"/>
<ElementType name="InvoiceCurrency" content="eltOnly" order="seq">
        <element type="Currency"/>
        <element type="AlternateCurrency" minOccurs="0" maxOccurs="1"/>
        <element type="Rate" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="TaxNumber" content="textOnly" dt:type="string"/>
<ElementType name="DDI" content="textOnly"/>
<ElementType name="Switchboard" content="textOnly"/>
<ElementType name="Fax" content="textOnly"/>
<ElementType name="Telex" content="textOnly"/>
<ElementType name="Email" content="textOnly"/>
<ElementType name="Mobile" content="textOnly"/>
<ElementType name="SpecialInstructions" content="textOnly" dt:type="string"/>
```

```
<ElementType name="DeliveryInformation" content="textOnly" dt:type="string"/>
<ElementType name="InvoiceLineInformation" content="textOnly" dt:type="string"/>
<ElementType name="ExtendedDescription" content="textOnly" dt:type="string"/>
<ElementType name="Properties" content="eltOnly" order="seq"><element
type="Quantity" minOccurs="0"
                    maxOccurs="1"/>
                    <element type="Length" minOccurs="0" maxOccurs="1"/>
                    <element type="Width" minOccurs="0" maxOccurs="1"/>
                    <element type="Depth" minOccurs="0" maxOccurs="1"/>
                    <element type="Weight" minOccurs="0" maxOccurs="1"/>
                    <element type="Volume" minOccurs="0" maxOccurs="1"/>
                    <element type="Height" minOccurs="0" maxOccurs="1"/>
                    <element type="Size" minOccurs="0" maxOccurs="1"/>
                    <element type="Color" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="SupplierReferences" content="eltOnly" order="seq">
          <element type="BuyersCodeForSupplier" minOccurs="0" maxOccurs="1"/>
          <element type="TaxNumber" minOccurs="0" maxOccurs="1"/>
          <element type="GLN" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="BuyersCodeForSupplier" content="textOnly"/>
<ElementType name="GLN" content="textOnly"/>
<ElementType name="BuyerReferences" content="eltOnly" order="seq">
          <element type="SuppliersCodeForBuyer" minOccurs="0" maxOccurs="1"/>
          <element type="TaxNumber" minOccurs="0" maxOccurs="1"/>
          <element type="GLN" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="SuppliersCodeForBuyer" content="textOnly"/>
<ElementType name="CarrierReferences" content="eltOnly" order="seq">
          <element type="BuyersCodeForCarrier" minOccurs="0" maxOccurs="1"/>
          <element type="GLN" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="BuyersCodeForCarrier" content="textOnly"/>
<ElementType name="Currency" content="textOnly" dt:type="string">
          <AttributeType name="Codelist" dt:type="string" default="ISO4217"/>
          <AttributeType name="Code" dt:type="string" required="yes"/>
                    <attribute type="Codelist"/>
                    <attribute type="Code"/>
</ElementType>
<ElementType name="Rate" content="textOnly" dt:type="float">
          <AttributeType name="CalculationType" dt:type="enumeration" dt:values="M
D" required="yes"/>
                    <attribute type="CalculationType"/>
</ElementType>
<ElementType name="Length" content="textOnly" dt:type="float">
          <AttributeType name="UOMDescription" dt:type="string"/>
          <AttributeType name="UOMCode" dt:type="string"/>
          <AttributeType name="UOMCodelist" dt:type="string" default="UNECE"/>
                    <attribute type="UOMDescription"/>
                    <attribute type="UOMCode"/>
                    <attribute type="UOMCodelist"/>
</ElementType>
<ElementType name="Width" content="textOnly" dt:type="float">
          <AttributeType name="UOMDescription" dt:type="string"/>
          <AttributeType name="UOMCode" dt:type="string"/>
          <AttributeType name="UOMCodelist" dt:type="string" default="UNECE"/>
                    <attribute type="UOMDescription"/>
```

```
                <attribute type="UOMCode"/>
                <attribute type="UOMCodelist"/>
</ElementType>
<ElementType name="Depth" content="textOnly" dt:type="float">
        <AttributeType name="UOMDescription" dt:type="string"/>
        <AttributeType name="UOMCode" dt:type="string"/>
        <AttributeType name="UOMCodelist" dt:type="string" default="UNECE"/>
                <attribute type="UOMDescription"/>
                <attribute type="UOMCode"/>
                <attribute type="UOMCodelist"/>
</ElementType>
<ElementType name="Weight" content="textOnly" dt:type="float">
        <AttributeType name="UOMDescription" dt:type="string"/>
        <AttributeType name="UOMCode" dt:type="string"/>
        <AttributeType name="UOMCodelist" dt:type="string" default="UNECE"/>
                <attribute type="UOMDescription"/>
                <attribute type="UOMCode"/>
                <attribute type="UOMCodelist"/>
</ElementType>
<ElementType name="Volume" content="textOnly" dt:type="float">
        <AttributeType name="UOMDescription" dt:type="string"/>
        <AttributeType name="UOMCode" dt:type="string"/>
        <AttributeType name="UOMCodelist" dt:type="string" default="UNECE"/>
                <attribute type="UOMDescription"/>
                <attribute type="UOMCode"/>
                <attribute type="UOMCodelist"/>
</ElementType>
<ElementType name="Height" content="textOnly" dt:type="float">
        <AttributeType name="UOMDescription" dt:type="string"/>
        <AttributeType name="UOMCode" dt:type="string"/>
        <AttributeType name="UOMCodelist" dt:type="string" default="UNECE"/>
                <attribute type="UOMDescription"/>
                <attribute type="UOMCode"/>
                <attribute type="UOMCodelist"/>
</ElementType>
<ElementType name="Size" content="textOnly" dt:type="float">
        <AttributeType name="UOMDescription" dt:type="string"/>
        <AttributeType name="UOMCode" dt:type="string"/>
        <AttributeType name="UOMCodelist" dt:type="string"/>
                <attribute type="UOMDescription"/>
                <attribute type="UOMCode"/>
                <attribute type="UOMCodelist"/>
</ElementType>
<ElementType name="Color" content="textOnly">
        <AttributeType name="UOMDescription" dt:type="string"/>
        <AttributeType name="UOMCode" dt:type="string"/>
        <AttributeType name="UOMCodelist" dt:type="string"/>
                <attribute type="UOMDescription"/>
                <attribute type="UOMCode"/>
                <attribute type="UOMCodelist"/>
</ElementType>
<ElementType name="Packsize" content="textOnly"/>
<ElementType name="BuyersProductCode" content="textOnly" dt:type="string"/>
<ElementType name="Amount" content="textOnly" dt:type="float"/>
<ElementType name="TradedUnitCode" content="textOnly"/>
<ElementType name="Units" content="textOnly"/>
<ElementType name="UnitPrice" content="textOnly" dt:type="float"/>
```

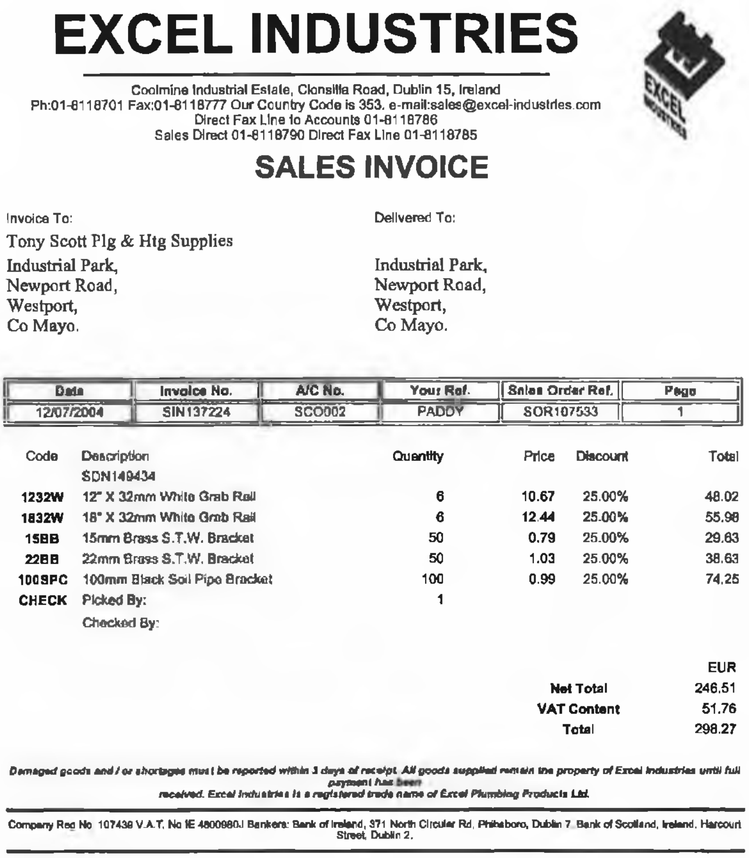
```
<ElementType name="SpecialPriceDescription" content="textOnly"/>
<ElementType name="Type" content="textOnly">
        <AttributeType name="Code" dt:type="string"/>
        <AttributeType name="Codelist" dt:type="string" default="BASDA"/>
                <attribute type="Code"/>
                <attribute type="Codelist"/>
</ElementType>
<ElementType name="Percentage" content="textOnly" dt:type="float"/>
<ElementType name="DeliverToReferences" content="eltOnly" order="seq">
        <element type="BuyersCodeForDelivery" minOccurs="0" maxOccurs="1"/>
        <element type="BuyersCodeForLocation" minOccurs="0" maxOccurs="1"/>
        <element type="GLN" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="BuyersCodeForDelivery" content="textOnly">
        <AttributeType name="Preserve" dt:type="enumeration" dt:values="true"
default="true"/>
                <attribute type="Preserve"/>
</ElementType>
<ElementType name="Location" content="textOnly"/>
<ElementType name="BuyersCodeForLocation" content="textOnly"/>
<ElementType name="BuyersOrderNumber" content="textOnly">
        <AttributeType name="Preserve" dt:type="enumeration" dt:values="true"
default="true"/>
                <attribute type="Preserve"/>
</ElementType>
<ElementType name="OrderLineNumber" content="textOnly"/>
<ElementType name="Settlement" content="eltOnly" order="seq">
        <element type="SettlementTerms" minOccurs="0" maxOccurs="1"/>
        <element type="SettlementMethod" minOccurs="0" maxOccurs="1"/>
        <element type="BankDetails" minOccurs="0" maxOccurs="1"/>
        <element type="CardDetails" minOccurs="0" maxOccurs="1"/>
        <element type="SettlementBy" minOccurs="0" maxOccurs="1"/>
        <element type="Contact" minOccurs="0" maxOccurs="*"/>
        <element type="SettlementDiscount" minOccurs="0" maxOccurs="*"/>
</ElementType>
<ElementType name="TaxSubTotal" content="eltOnly" order="seq">
        <element type="TaxRate"/>
        <element type="NumberOfLinesAtRate"/>
        <element type="TotalValueAtRate"/>
        <element type="SettlementDiscountAtRate" minOccurs="0" maxOccurs="1"/>
        <element type="TaxableValueAtRate"/>
        <element type="TaxAtRate"/>
        <element type="NetPaymentAtRate"/>
        <element type="GrossPaymentAtRate"/>
        <element type="TaxCurrency"/>
</ElementType>
<ElementType name="TaxPointDate" content="textOnly"/>
<ElementType name="SettlementTerms" content="eltOnly" order="one">
        <AttributeType name="Code" dt:type="string"/>
        <AttributeType name="Codelist" dt:type="string" default="BASDA"/>
                <attribute type="Code"/>
                <attribute type="Codelist"/>
                <element type="PayByDate"/>
        <element type="DaysFromInvoice" minOccurs="0" maxOccurs="1"/>
        <element type="DaysFromMonthEnd" minOccurs="0" maxOccurs="1"/>
        <element type="DaysFromDelivery" minOccurs="0" maxOccurs="1"/>
        <element type="PaidDate" minOccurs="0" maxOccurs="1"/>
        <element type="Description" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="SettlementMethod" content="textOnly">
        <AttributeType name="Code" dt:type="string"/>
        <AttributeType name="Codelist" dt:type="string" default="BASDA"/>
                <attribute type="Code"/>
                <attribute type="Codelist"/>
</ElementType>
<ElementType name="BankDetails" content="eltOnly order="seq">
        <element type="BankCode"/>
        <element type="BankReference"/>
        <element type="GLN" minOccurs="0" maxOccurs="1"/>
        <element type="Address" minOccurs="0" maxOccurs="1"/>
        <element type="Contact" minOccurs="0" maxOccurs="*"/>
</ElementType>
<ElementType name="SettlementBy" content="eltOnly" order="seq">
        <element type="SettlementByReferences" minOccurs="0" maxOccurs="1"/>
        <element type="Party" minOccurs="1" maxOccurs="*"/>
        <element type="Address" minOccurs="0" maxOccurs="1"/>
        <element type="Contact" minOccurs="0" maxOccurs="*"/>
</ElementType>
<ElementType name="SettlementDiscount" content="eltOnly" order="seq">
        <AttributeType name="Code" dt:type="string"/>
        <AttributeType name="Codelist" dt:type="string" default="BASDA"/>
                <attribute type="Code"/>
                <attribute type="Codelist"/>
                <element type="PercentDiscount" minOccurs="0" maxOccurs="1"/>
                <element type="AmountDiscount"/>
</ElementType>
<ElementType name="BankCode" content="textOnly"/>
<ElementType name="BankReference" content="textOnly"/>
<ElementType name="NumberOfLinesAtRate" content="textOnly" dt:type="int"/>
<ElementType name="TotalValueAtRate" content="textOnly" dt:type="float"/>
<ElementType name="SettlementDiscountAtRate" content="textOnly" dt:type="float"/>
<ElementType name="TaxableValueAtRate" content="textOnly" dt:type="float"/>
<ElementType name="NetPaymentAtRate" content="textOnly" dt:type="float"/>
<ElementType name="GrossPaymentAtRate" content="textOnly" dt:type="float"/>
<ElementType name="AlternateCurrency" content="textOnly">
        <AttributeType name="Code" dt:type="string" required="yes"/>
        <AttributeType name="Codelist" dt:type="string" default="ISO4217"/>
                <attribute type="Code"/>
                <attribute type="Codelist"/>
</ElementType>
<ElementType name="TaxCurrency" content="eltOnly" order="seq">
        <element type="Currency"/>
        <element type="AlternateCurrency" minOccurs="0" maxOccurs="1"/>
        <element type="Rate" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="LotSerial" content="eltOnly" order="seq">
        <element type="LotSerialType"/>
        <element type="LotSerialNumber"/>
        <element type="Quantity" minOccurs="0" maxOccurs="1"/>
        <element type="LotSerialDates" minOccurs="0" maxOccurs="*"/>
        <element type="Narrative" minOccurs="0" maxOccurs="*"/>
</ElementType>
<ElementType name="LotSerialType" content="textOnly">
        <AttributeType name="Code" dt:type="string" required="yes"/>
        <AttributeType name="Codelist" dt:type="string" default="BASDA"/>
                <attribute type="Code"/>
        <attribute type="Codelist"/>
</ElementType>
<ElementType name="LotSerialNumber" content="textOnly"/>
<ElementType name="LotSerialDates" content="eltOnly" order="seq">
        <element type="BestBeforeDate" minOccurs="0" maxOccurs="1"/>
        <element type="DeliveryDate" minOccurs="0" maxOccurs="1"/>
        <element type="DespatchDate" minOccurs="0" maxOccurs="1"/>
        <element type="OnSaleByDate" minOccurs="0" maxOccurs="1"/>
        <element type="SellByDate" minOccurs="0" maxOccurs="1"/>
        <element type="ShelfLifeDate" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="BestBeforeDate" content="textOnly" dt:type="dateTime"/>
<ElementType name="DeliveryDate" content="textOnly" dt:type="dateTime"/>
<ElementType name="DespatchDate" content="textOnly" dt:type="dateTime"/>
<ElementType name="OnSaleByDate" content="textOnly" dt:type="dateTime"/>
<ElementType name="SellByDate" content="textOnly" dt:type="dateTime"/>
<ElementType name="ShelfLifeDate" content="textOnly" dt:type="dateTime"/>
<ElementType name="CardDetails" content="eltOnly" order="seq">
        <AttributeType name="CardType" dt:type="enumeration" dt:values="Credit
Debit Charge"
        required="yes"/>
                <attribute type="CardType"/>
                <element type="Issuer"/>
                <element type="Party"/>
                <element type="Address" minOccurs="0" maxOccurs="1"/>
                <element type="IssueDate" minOccurs="0" maxOccurs="1"/>
                <element type="ExpiryDate"/>
                <element type="IssueNumber" minOccurs="0" maxOccurs="1"/>
                <element type="CardNumber"/>
                <element type="AuthorisationCode" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="Issuer" content="textOnly"/>
<ElementType name="IssueDate" content="textOnly" dt:type="dateTime"/>
<ElementType name="ExpiryDate" content="textOnly" dt:type="dateTime"/>
<ElementType name="IssueNumber" content="textOnly" dt:type="int"/>
<ElementType name="CardNumber" content="textOnly"/>
<ElementType name="ConsumerUnitCode" content="textOnly"/>
<ElementType name="NumberOfLines" content="textOnly" dt:type="int"/>
<ElementType name="NumberOfTaxRates" content="textOnly" dt:type="int"/>
<ElementType name="LineValueTotal" content="textOnly" dt:type="float"/>
<ElementType name="NetPaymentTotal" content="textOnly" dt:type="float"/>
<ElementType name="GrossPaymentTotal" content="textOnly" dt:type="float"/>
<ElementType name="SuppliersInvoiceNumber" content="textOnly">
        <AttributeType name="Preserve" dt:type="enumeration" dt:values="true"
default="true"/>
                <attribute type="Preserve"/>
</ElementType>
<ElementType name="TaxAtRate" content="textOnly"/>
<ElementType name="SettlementDiscountTotal" content="textOnly" dt:type="float"/>
<ElementType name="TaxableTotal" content="textOnly" dt:type="float"/>
<ElementType name="OriginatingSoftware" content="eltOnly" order="seq">
        <element type="SoftwareManufacturer"/>
        <element type="SoftwareProduct"/>
        <element type="SoftwareVersion"/>
</ElementType>
```

```
<ElementType name="TestFlag" content="mixed" order="many">
        <AttributeType name="TestReference" dt:type="string"/>
                <attribute type="TestReference"/>
                <element type="Test"/>
</ElementType>
<ElementType name="SoftwareManufacturer" content="textOnly"/>
<ElementType name="SoftwareVersion" content="textOnly"/>
<ElementType name="SoftwareProduct" content="textOnly"/>
<ElementType name="Schema" content="eltOnly" order="seq">
        <element type="Version"/>
</ElementType>
<ElementType name="Stylesheet" content="eltOnly" order="seq">
        <element type="StylesheetOwner"/>
        <element type="StylesheetName"/>
        <element type="Version"/>
        <element type="StylesheetType"/>
</ElementType>
<ElementType name="Parameters" content="eltOnly" order="seq">
        <element type="Language"/>
        <element type="DecimalSeparator"/>
        <element type="Precision"/>
</ElementType>
<ElementType name="Version" content="textOnly"/>
<ElementType name="StylesheetOwner" content="textOnly"/>
<ElementType name="StylesheetName" content="textOnly"/>
<ElementType name="StylesheetType" content="textOnly"/>
<ElementType name="Language" content="textOnly"/>
<ElementType name="DecimalSeparator" content="textOnly" dt:type="char"/>
<ElementType name="Precision" content="textOnly" dt:type="float"/>
<ElementType name="AuthorisationCode" content="textOnly"/>
<ElementType name="BuyersOrderLineReference" content="textOnly">
        <AttributeType name="Preserve" dt:type="enumeration" dt:values="true"
default="true"/>
                <attribute type="Preserve"/>
</ElementType>
<ElementType name="InvoiceToReferences" content="eltOnly" order="seq">
        <element type="GLN" minOccurs="0" maxOccurs="1"/>
        <element type="SuppliersCodeForInvoiceTo" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="SettlementByReferences" content="eltOnly" order="seq">
        <element type="GLN" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="Test" content="empty">
        <AttributeType name="ExpectedReply" dt:type="string"/>
        <AttributeType name="Mode" dt:type="string" required="yes"/>
                <attribute type="ExpectedReply"/>
                <attribute type="Mode"/>
</ElementType>
<ElementType name="TypeOfSupply" content="textOnly" dt:type="string"/>
<ElementType name="Intrastat" content="eltOnly" order="seq">
        <element type="CommodityCode" minOccurs="0" maxOccurs="1"/>
        <element type="CommodityDescription" minOccurs="0" maxOccurs="1"/>
        <element type="NatureOfTransaction" minOccurs="0" maxOccurs="1"/>
        <element type="SupplementaryUnits" minOccurs="0" maxOccurs="1"/>
        <element type="CountryOfDestination" minOccurs="0" maxOccurs="1"/>
        <element type="ModeOfTransport" minOccurs="0" maxOccurs="1"/>
</ElementType>
```

```
<ElementType name="CommodityCode" content="textOnly" dt:type="string"/>
<ElementType name="CommodityDescription" content="textOnly" dt:type="string"/>
<ElementType name="NatureOfTransaction" content="textOnly" dt:type="string">
        <AttributeType name="Code" dt:type="string" default="10"/>
        <AttributeType name="Codelist" dt:type="string" default="BASDA"/>
                <attribute type="Code"/>
                <attribute type="Codelist"/>
</ElementType>
<ElementType name="SupplementaryUnits" content="textOnly" dt:type="int"/>
<ElementType name="CountryOfDestination" content="textOnly" dt:type="string">
        AttributeType name="Code" dt:type="string" default="GB"/>
        <AttributeType name="Codelist" dt:type="string" default="BASDA"/>
                <attribute type="Code"/>
                <attribute type="Codelist"/>
</ElementType>
<ElementType name="ModeOfTransport" content="textOnly" dt:type="string">
        AttributeType name="Code" dt:type="string" default="1"/>
        <AttributeType name="Codelist" dt:type="string" default="BASDA"/>
                <attribute type="Code"/>
                <attribute type="Codelist"/>
</ElementType>
<ElementType name="SuppliersCodeForInvoiceTo" content="textOnly"/>
<ElementType name="PayByDate" content="textOnly"/>
<ElementType name="DaysFromInvoice" content="textOnly"/>
<ElementType name="DaysFromMonthEnd" content="textOnly"/>
<ElementType name="DaysFromDelivery" content="textOnly"/>
<ElementType name="QualifyingTerms" content="eltOnly" order="one">
        <AttributeType name="Code" dt:type="string"/>
        <AttributeType name="Codelist" dt:type="string" default="BASDA"/>
                <attribute type="Code"/>
                <attribute type="Codelist"/>
                <element type="PayByDate"/>
                <element type="DaysFromInvoice" minOccurs="0" maxOccurs="1"/>
                <element type="DaysFromMonthEnd" minOccurs="0"
maxOccurs="1"/>
                <element type="DaysFromDelivery" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="PaidDate" content="textOnly" dt:type="date"/>
</Schema>
```

Appendix 3

Sample XML Code



EXCEL INDUSTRIES

Coolmine Industrial Estate, Clonsilla Road, Dublin 15, Ireland
Ph:01-8118701 Fax:01-8118777 Our Country Code is 353. e-mail:sales@excel-industries.com
Direct Fax Line to Accounts 01-8118786
Sales Direct 01-8118790 Direct Fax Line 01-8118785

SALES INVOICE

Invoice To:

Tony Scott Plg & Htg Supplies
Industrial Park,
Newport Road,
Westport,
Co Mayo.

Delivered To:

Industrial Park,
Newport Road,
Westport,
Co Mayo.

Date	Invoice No.	A/C No.	Your Ref.	Sales Order Ref.	Page
12/07/2004	SIN137224	SCO002	PADDY	SOR107533	1

Code	Description	Quantity	Price	Discount	Total
	SDN149434				
1232W	12" X 32mm White Grab Rail	6	10.67	25.00%	48.02
1832W	18" X 32mm White Grab Rail	6	12.44	25.00%	55.98
15BB	15mm Brass S.T.W. Bracket	50	0.79	25.00%	29.63
22BB	22mm Brass S.T.W. Bracket	50	1.03	25.00%	38.63
100SPC	100mm Black Soil Pipe Bracket	100	0.99	25.00%	74.25
CHECK	Picked By:	1			
	Checked By:				

	EUR
Net Total	246.51
VAT Content	51.76
Total	298.27

Damaged goods and / or shortages must be reported within 3 days of receipt. All goods supplied remain the property of Excel Industries until full payment has been received. Excel Industries is a registered trade name of Excel Plumbing Products Ltd.

Company Reg No 107439 V.A.T. No IE 4800980J Bankers: Bank of Ireland, 371 North Circular Rd, Phibsboro, Dublin 7. Bank of Scotland, Ireland, Harcourt Street, Dublin 2.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="excheqr.xsl"?>
<biztalk_1 xmlns="urn:schemas-biztalk-org:biztalk/biztalk_1.xml">
 <header>
  <delivery>
   <message>
    <messageID/>
    <sent>2004-12-07T12:37:05</sent>
    <subject>Sales Invoice</subject>
    <basda:SENDER xmlns:basda="urn:basda.org:header">Excel Industries -
**Euro**</basda:SENDER>
    <basda:RECIPIENT xmlns:basda="urn:basda.org:header">Tony Scott Plg &amp; Htg
Supplies</basda:RECIPIENT>
   </message>
   <to>
    <address/>
    <state>
     <referenceID/>
     <handle/>
     <process/>
    </state>
   </to>
   <from>
    <address/>
    <state>
     <referenceID/>
     <handle/>
     <process/>
    </state>
   </from>
  </delivery>
  <manifest/>
 </header>
 <body>
  <Invoice xmlns="urn:schemas-basda-org:2000:salesInvoice:xdr:3.01"
xmlns:exchequer="urn:www.exchequer.com">
   <InvoiceHead>
    <Schema>
     <Version>3.01</Version>
    </Schema>
    <Stylesheet>
     <StylesheetOwner>BASDA</StylesheetOwner>
     <StylesheetName>eBIS-XML_simple.xsl</StylesheetName>
     <Version>3.0</Version>
     <StylesheetType>xsl</StylesheetType>
    </Stylesheet>
    <Parameters>
     <Language>en_GB</Language>
     <DecimalSeparator>.</DecimalSeparator>
     <Precision>20.3</Precision>
    </Parameters>
    <OriginatingSoftware>
     <SoftwareManufacturer>Exchequer Software Ltd</SoftwareManufacturer>
     <SoftwareProduct>Enterprise</SoftwareProduct>
     <SoftwareVersion>5.52</SoftwareVersion>
    </OriginatingSoftware>
    <InvoiceType Code="INV">Sales Invoice</InvoiceType>
    <InvoiceCurrency>
     <Currency Code="EUR">Euro</Currency>
    </InvoiceCurrency>
    <Intrastat>
     <CommodityCode/>
     <CommodityDescription/>
     <NatureOfTransaction>1</NatureOfTransaction>
     <SupplementaryUnits/>
     <CountryOfDestination/>
     <ModeOfTransport>1</ModeOfTransport>
    </Intrastat>
    <Checksum>88004</Checksum>
   </InvoiceHead>
   <InvoiceReferences>
    <BuyersOrderNumber Preserve="true">PADDY</BuyersOrderNumber>
    <SuppliersInvoiceNumber Preserve="true">SIN137224</SuppliersInvoiceNumber>
    <exchequer:YourRef>SOR107533</exchequer:YourRef>
    <exchequer:HeaderUserDef3>RB</exchequer:HeaderUserDef3>
   </InvoiceReferences>
   <InvoiceDate>2004-07-12</InvoiceDate>
   <Supplier>
    <SupplierReferences>
     <BuyersCodeForSupplier/>
     <TaxNumber>IE 4800980J</TaxNumber>
    </SupplierReferences>
    <Party>Excel Industries - **Euro**</Party>
    <Address/>
```

```
</Supplier>
<Buyer>
  <BuyerReferences>
    <SuppliersCodeForBuyer>SCO002</SuppliersCodeForBuyer>
  </BuyerReferences>
  <Party>Tony Scott Plg &amp; Htg Supplies</Party>
  <Address>
    <AddressLine>Industrial Park,</AddressLine>
    <AddressLine>Newport Road,</AddressLine>
    <AddressLine>Westport,</AddressLine>
    <AddressLine>Co Mayo.</AddressLine>
  </Address>
  <Contact>
    <Name>Paddy Welsh</Name>
    <DDI>09825442 / 28885</DDI>
    <Fax>09828103</Fax>
  </Contact>
</Buyer>
<Delivery>
  <DeliverTo>
    <Party>Tony Scott Plg &amp; Htg Supplies</Party>
    <Address>
      <AddressLine>Industrial Park,</AddressLine>
      <AddressLine>Newport Road,</AddressLine>
      <AddressLine>Westport,</AddressLine>
      <AddressLine>Co Mayo.</AddressLine>
    </Address>
  </DeliverTo>
  <LatestAcceptableDate>2004-09-30</LatestAcceptableDate>
</Delivery>
<InvoiceTo>
  <InvoiceToReferences>
    <exchequer:SuppliersCodeForInvoiceTo>SCO002</exchequer:SuppliersCodeForInvoiceTo>
    <SuppliersCodeForInvoiceTo>SCO002</SuppliersCodeForInvoiceTo>
  </InvoiceToReferences>
  <Party>Tony Scott Plg &amp; Htg Supplies</Party>
  <Address>
    <AddressLine>Industrial Park,</AddressLine>
    <AddressLine>Newport Road,</AddressLine>
    <AddressLine>Westport,</AddressLine>
    <AddressLine>Co Mayo.</AddressLine>
  </Address>
  <Contact>
    <Name>Paddy Welsh</Name>
    <DDI>09825442 / 28885</DDI>
    <Fax>09828103</Fax>
  </Contact>
</InvoiceTo>
<InvoiceLine Action="Add" TypeCode="DES" TypeDescription="Description">
  <LineNumber Preserve="true">1</LineNumber>
  <Product>
    <Description>SDN149434</Description>
  </Product>
</InvoiceLine>
<InvoiceLine Action="Add" TypeCode="INL" TypeDescription="Invoice line">
  <LineNumber Preserve="true">2</LineNumber>
  <InvoiceLineReferences>
```

```
    <CostCentre>DIS</CostCentre>
    <exchequer:Department>11 </exchequer:Department>
    <GeneralLedgerCode>52010</GeneralLedgerCode>
    <OrderLineNumber>1</OrderLineNumber>
    <BuyersOrderLineReference Preserve="true">SIN137224-2</BuyersOrderLineReference>
  </InvoiceLineReferences>
  <Product>
    <SuppliersProductCode>1232W</SuppliersProductCode>
    <ConsumerUnitCode>539126960117</ConsumerUnitCode>
    <Description>12&quot; X 32mm White Grab Rail</Description>
    <Properties>
      <Weight UOMDescription="kilogrammes" UOMCode="KGM">0.50</Weight>
    </Properties>
  </Product>
  <Quantity UOMDescription="1">
    <Packsize>1.00</Packsize>
    <Amount>6.00</Amount>
  </Quantity>
  <Price UOMDescription="1">
    <UnitPrice>10.6700</UnitPrice>
  </Price>
  <PercentDiscount>
    <Type Code="RED">Derived discount</Type>
    <Percentage>25.0000</Percentage>
  </PercentDiscount>
  <LineTax>
    <TaxRate Code="S">21.00</TaxRate>
    <TaxValue>10.080</TaxValue>
  </LineTax>
  <LineTotal>48.0200</LineTotal>
  <Intrastat>
    <CommodityCode>73089099</CommodityCode>
    <CommodityDescription/>
    <NatureOfTransaction>1</NatureOfTransaction>
    <SupplementaryUnits>1</SupplementaryUnits>
    <CountryOfDestination/>
    <ModeOfTransport>1</ModeOfTransport>
  </Intrastat>
</InvoiceLine>
<InvoiceLine Action="Add" TypeCode="INL" TypeDescription="Invoice line">
  <LineNumber Preserve="true">3</LineNumber>
  <InvoiceLineReferences>
    <CostCentre>DIS</CostCentre>
    <exchequer:Department>11 </exchequer:Department>
    <GeneralLedgerCode>52010</GeneralLedgerCode>
    <OrderLineNumber>2</OrderLineNumber>
    <BuyersOrderLineReference Preserve="true">SIN137224-3</BuyersOrderLineReference>
  </InvoiceLineReferences>
  <Product>
    <SuppliersProductCode>1832W</SuppliersProductCode>
    <ConsumerUnitCode>539126960165</ConsumerUnitCode>
    <Description>18&quot; X 32mm White Grab Rail</Description>
    <Properties>
      <Weight UOMDescription="kilogrammes" UOMCode="KGM">0.55</Weight>
    </Properties>
  </Product>
  <Quantity UOMDescription="1">
```

```
    <Packsize>1.00</Packsize>
    <Amount>6.00</Amount>
  </Quantity>
  <Price UOMDescription="1">
    <UnitPrice>12.4400</UnitPrice>
  </Price>
  <PercentDiscount>
    <Type Code="RED">Derived discount</Type>
    <Percentage>25.0000</Percentage>
  </PercentDiscount>
  <LineTax>
    <TaxRate Code="S">21.00</TaxRate>
    <TaxValue>11.760</TaxValue>
  </LineTax>
  <LineTotal>55.9800</LineTotal>
  <Intrastat>
    <CommodityCode>73089099</CommodityCode>
    <CommodityDescription/>
    <NatureOfTransaction>1</NatureOfTransaction>
    <SupplementaryUnits>1</SupplementaryUnits>
    <CountryOfDestination/>
    <ModeOfTransport>1</ModeOfTransport>
  </Intrastat>
</InvoiceLine>
<InvoiceLine Action="Add" TypeCode="INL" TypeDescription="Invoice line">
  <LineNumber Preserve="true">4</LineNumber>
  <InvoiceLineReferences>
    <CostCentre>DIS</CostCentre>
    <exchequer:Department>11 </exchequer:Department>
    <GeneralLedgerCode>52010</GeneralLedgerCode>
    <OrderLineNumber>3</OrderLineNumber>
    <BuyersOrderLineReference Preserve="true">SIN137224-4</BuyersOrderLineReference>
  </InvoiceLineReferences>
  <Product>
    <SuppliersProductCode>15BB</SuppliersProductCode>
    <ConsumerUnitCode>539126920221</ConsumerUnitCode>
    <Description>15mm Brass S.T.W. Bracket</Description>
    <Properties>
      <Weight UOMDescription="kilogrammes" UOMCode="KGM">0.02</Weight>
    </Properties>
  </Product>
  <Quantity UOMDescription="each">
    <Packsize>1.00</Packsize>
    <Amount>50.00</Amount>
  </Quantity>
  <Price UOMDescription="each">
    <UnitPrice>0.7900</UnitPrice>
  </Price>
  <PercentDiscount>
    <Type Code="RED">Derived discount</Type>
    <Percentage>25.0000</Percentage>
  </PercentDiscount>
  <LineTax>
    <TaxRate Code="S">21.00</TaxRate>
    <TaxValue>6.220</TaxValue>
  </LineTax>
  <LineTotal>29.6300</LineTotal>
  <Intrastat>
    <CommodityCode>83024100</CommodityCode>
    <CommodityDescription/>
    <NatureOfTransaction>1</NatureOfTransaction>
    <SupplementaryUnits>1</SupplementaryUnits>
    <CountryOfDestination/>
    <ModeOfTransport>1</ModeOfTransport>
  </Intrastat>
</InvoiceLine>
<InvoiceLine Action="Add" TypeCode="INL" TypeDescription="Invoice line">
  <LineNumber Preserve="true">5</LineNumber>
  <InvoiceLineReferences>
    <CostCentre>DIS</CostCentre>
    <exchequer:Department>11 </exchequer:Department>
    <GeneralLedgerCode>52010</GeneralLedgerCode>
    <OrderLineNumber>4</OrderLineNumber>
    <BuyersOrderLineReference Preserve="true">SIN137224-5</BuyersOrderLineReference>
  </InvoiceLineReferences>
  <Product>
    <SuppliersProductCode>22BB</SuppliersProductCode>
    <ConsumerUnitCode>539126920220</ConsumerUnitCode>
    <Description>22mm Brass S.T.W. Bracket</Description>
    <Properties>
      <Weight UOMDescription="kilogrammes" UOMCode="KGM">0.04</Weight>
    </Properties>
  </Product>
  <Quantity UOMDescription="each">
    <Packsize>1.00</Packsize>
    <Amount>50.00</Amount>
  </Quantity>
  <Price UOMDescription="each">
    <UnitPrice>1.0300</UnitPrice>
  </Price>
  <PercentDiscount>
    <Type Code="RED">Derived discount</Type>
    <Percentage>25.0000</Percentage>
  </PercentDiscount>
  <LineTax>
    <TaxRate Code="S">21.00</TaxRate>
    <TaxValue>8.110</TaxValue>
  </LineTax>
  <LineTotal>38.6300</LineTotal>
  <Intrastat>
    <CommodityCode>83024100</CommodityCode>
    <CommodityDescription/>
    <NatureOfTransaction>1</NatureOfTransaction>
    <SupplementaryUnits>1</SupplementaryUnits>
    <CountryOfDestination/>
    <ModeOfTransport>1</ModeOfTransport>
  </Intrastat>
</InvoiceLine>
<InvoiceLine Action="Add" TypeCode="INL" TypeDescription="Invoice line">
  <LineNumber Preserve="true">6</LineNumber>
  <InvoiceLineReferences>
    <CostCentre>DIS</CostCentre>
    <exchequer:Department>11 </exchequer:Department>
    <GeneralLedgerCode>52010</GeneralLedgerCode>
    <OrderLineNumber>5</OrderLineNumber>
    <BuyersOrderLineReference Preserve="true">SIN137224-6</BuyersOrderLineReference>
  </InvoiceLineReferences>
  <Product>
    <SuppliersProductCode>100SPC</SuppliersProductCode>
    <ConsumerUnitCode>539126954292</ConsumerUnitCode>
    <Description>100mm Black Soil Pipe Bracket</Description>
    <Properties>
      <Weight UOMDescription="kilogrammes" UOMCode="KGM">0.04</Weight>
    </Properties>
  </Product>
  <Quantity UOMDescription="each">
    <Packsize>1.00</Packsize>
    <Amount>100.00</Amount>
  </Quantity>
  <Price UOMDescription="each">
    <UnitPrice>0.9900</UnitPrice>
  </Price>
  <PercentDiscount>
    <Type Code="RED">Derived discount</Type>
    <Percentage>25.0000</Percentage>
  </PercentDiscount>
  <LineTax>
    <TaxRate Code="S">21.00</TaxRate>
    <TaxValue>15.590</TaxValue>
  </LineTax>
  <LineTotal>74.2500</LineTotal>
  <Intrastat>
    <CommodityCode>39259010</CommodityCode>
    <CommodityDescription/>
    <NatureOfTransaction>1</NatureOfTransaction>
    <SupplementaryUnits>1</SupplementaryUnits>
    <CountryOfDestination/>
    <ModeOfTransport>1</ModeOfTransport>
  </Intrastat>
</InvoiceLine>
<InvoiceLine Action="Add" TypeCode="INL" TypeDescription="Invoice line">
  <LineNumber Preserve="true">7</LineNumber>
  <InvoiceLineReferences>
    <GeneralLedgerCode>52010</GeneralLedgerCode>
    <OrderLineNumber>6</OrderLineNumber>
    <BuyersOrderLineReference Preserve="true">SIN137224-7</BuyersOrderLineReference>
  </InvoiceLineReferences>
  <Product>
    <SuppliersProductCode>CHECK</SuppliersProductCode>
    <Description>Picked By:</Description>
  </Product>
  <Quantity UOMDescription="each">
    <Packsize>1.00</Packsize>
    <Amount>1.00</Amount>
  </Quantity>
  <Price UOMDescription="each">
    <UnitPrice>0.0000</UnitPrice>
  </Price>
  <PercentDiscount>
    <Type Code="RED">Derived discount</Type>
    <Percentage>25.0000</Percentage>
  </PercentDiscount>
  <LineTax>
    <TaxRate Code="S">21.00</TaxRate>
    <TaxValue>0.000</TaxValue>
  </LineTax>
  <LineTotal>0.0000</LineTotal>
  <Intrastat>
    <CommodityCode/>
    <CommodityDescription/>
    <NatureOfTransaction>1</NatureOfTransaction>
    <SupplementaryUnits>1</SupplementaryUnits>
    <CountryOfDestination/>
    <ModeOfTransport>1</ModeOfTransport>
  </Intrastat>
</InvoiceLine>
<InvoiceLine Action="Add" TypeCode="DES" TypeDescription="Description">
  <LineNumber Preserve="true">8</LineNumber>
  <Product>
    <Description/>
  </Product>
</InvoiceLine>
<InvoiceLine Action="Add" TypeCode="DES" TypeDescription="Description">
  <LineNumber Preserve="true">9</LineNumber>
  <Product>
    <Description>Checked By:</Description>
  </Product>
</InvoiceLine>
<Narrative>On Statement.</Narrative>
<Narrative>On Statement.</Narrative>
<Narrative>On Statement.</Narrative>
<Narrative>On Statement.</Narrative>
<Narrative>On Statement.</Narrative>
<Settlement>
  <SettlementTerms>
    <PayByDate>2004-09-30</PayByDate>
  </SettlementTerms>
  <SettlementDiscount>
    <PercentDiscount>
      <QualifyingTerms>
        <PayByDate>2004-09-10</PayByDate>
        <DaysFromInvoice>60</DaysFromInvoice>
      </QualifyingTerms>
      <Percentage>0.00</Percentage>
    </PercentDiscount>
    <AmountDiscount>
      <QualifyingTerms>
        <PayByDate>2004-09-10</PayByDate>
        <DaysFromInvoice>60</DaysFromInvoice>
      </QualifyingTerms>
      <Amount>0.00</Amount>
    </AmountDiscount>
  </SettlementDiscount>
</Settlement>
<TaxSubTotal>
  <TaxRate Code="S">21.00</TaxRate>
  <NumberOfLinesAtRate>6</NumberOfLinesAtRate>
  <TotalValueAtRate>246.51</TotalValueAtRate>
```

```
        <SettlementDiscountAtRate>0.00</SettlementDiscountAtRate>
        <TaxableValueAtRate>246.51</TaxableValueAtRate>
        <TaxAtRate>51.76</TaxAtRate>
        <NetPaymentAtRate>298.27</NetPaymentAtRate>
        <GrossPaymentAtRate>298.27</GrossPaymentAtRate>
        <TaxCurrency>
          <Currency Code="EUR">Euro</Currency>
        </TaxCurrency>
      </TaxSubTotal>
      <InvoiceTotal>
        <NumberOfLines>9</NumberOfLines>
        <NumberOfTaxRates>1</NumberOfTaxRates>
        <LineValueTotal>246.5100</LineValueTotal>
        <SettlementDiscountTotal>0.0000</SettlementDiscountTotal>
        <TaxableTotal>246.5100</TaxableTotal>
        <TaxTotal>51.7600</TaxTotal>
        <NetPaymentTotal>298.2700</NetPaymentTotal>
        <GrossPaymentTotal>298.2700</GrossPaymentTotal>
      </InvoiceTotal>
    </Invoice>
  </body>
</biztalk_1>
```

Appendix 4

Stylesheet Coding

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
:basda="urn:schemas-basda-org:2000:salesInvoice:xdr:3.01" xmlns:biztalk=
"urn:schemas-biztalk-org:biztalk/biztalk_1.xml" xmlns:exchequer="urn:www.exchequer.com"
:v="urn:schemas-microsoft-com:vml">
  <xsl:output method="html" encoding="iso-8859-1"/>
  <xsl:decimal-format NaN=""/>
  <xsl:strip-space elements="*"/>
  <xsl:variable name="creditnote">
    <!-- Check if this is a Credit Note -->
    <xsl:choose>
      <xsl:when test="/biztalk:biztalk_1/biztalk:body/basda:Invoice/basda:InvoiceHead/basda:
    = 'Sales Invoice'">
        <xsl:text>false</xsl:text>
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>true</xsl:text>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:variable name="usebarcode">
    <!-- Check forGroup Customer  Type 1 -->
    <xsl:choose>
      <xsl:when test="starts-
with(/biztalk:biztalk_1/biztalk:body/basda:Invoice/basda:Buyer/basda:
    /basda:SuppliersCodeForBuyer,'B&amp;Q')">
        <xsl:text>true</xsl:text>
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>false</xsl:text>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:variable name="swapaddresses">
    <!-- Check for Group Customer Type 2 -->
    <xsl:choose>
      <xsl:when test="/biztalk:biztalk_1/biztalk:body/basda:Invoice/basda:Buyer/basda:
    /basda:SuppliersCodeForBuyer = 'CHA0'">
        <xsl:text>true</xsl:text>
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>false</xsl:text>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:template match="/">
    <html>
      <head>
        <title>Excel Industries Ltd.
          <xsl:value-of select="/biztalk:biztalk_1/biztalk:body/basda:Invoice/basda:
        /basda:InvoiceType"/>
        </title>
        <style>
    \:* { behavior: url(#default#VML); }
       .rhdr {color:navy; font-family:arial; font-size:36pt; font-weight:bold; margin-
bottom:0px;
      bottom:0px;}
       .desc {color:navy; font-family:arial; font-size:9pt; margin-top:0px;}
      {color:black; font-family:arial; font-size:20pt; font-weight:bold;margin-bottom:0px;}
      {color:red; font-family:arial; font-size:20pt; font-weight:bold;}
       .addr {color:black; font-family:arial; font-size:11pt; font-weight:bold;}
        .ftr {color:blck; font-family:arial; font-size:7pt; text-align=center; font-weight:bold;
       font-style:italic;
      bottom:0px; padding-bottom:0px;}
       .bftr {color:navy; font-family:arial; font-size:7pt; text-align=center;margin-bottom:0px;
      bottom:0px;margin-top:0px;}
       .hdr {color:black; font-family:arial; font-size:9pt; font-weight:bold;}
       .std {color:black; font-family:arial; font-size:9pt;margin-top:0px;}
       .curr {color:black; font-family:arial; font-size:9pt;}
       .tot {color:black; font-family:arial; font-size:9pt;}
       .btot {color:black; font-family:arial; font-size:9pt; font-weight:bold;}
       .code {color:black; font-family:arial; font-size:9pt; font-weight:bold;}
    </style>
   </head>
   <body bgcolor="#FFFFFF" text="#000000">
    <table width="100%" border="0">
     <tr>
      <td width="85%" align="center" valign="top">
       <p class="rhdr">INDUSTRIES </p>
       <hr style="color:red" width="75%"/>
       <p class="desc">Coolmine Industrial Estate, Clonsilla Road, Dublin 15, Ireland
        <br/>
       :01-8118701 Fax:01-8118777 Our Country Code is 353. e-mail:
       @excel-industries.com
        <br/>
       Fax Line to Accounts 01-8118786
        <br/>
       Direct 01-8118790 Direct Fax Line 01-8118785
       </p>
      </td>
      <td>  <img src="logo4.png" height="121" width="93"/></td>
     </tr>
    </table>
    <table width="100%" border="0">
     <tr>
      <td width="100%" align="center">
       <xsl:if test="$creditnote='false'">
        <H2>SALES INVOICE</H2>
       </xsl:if>
       <xsl:if test="$creditnote='true'">
        <H3>CREDIT NOTE</H3>
       </xsl:if>
       <br/>
      </td>
     </tr>
    </table>
    <xsl:apply-templates select="/biztalk:biztalk_1/biztalk:body/basda:Invoice"/>
    <p class="ftr">
   goods and / or shortages must be reported within 3 days of receipt.
```

```
    goods supplied remain the property of Excel Industries until full payment has been
      <br/>
    . Excel Industries is a registered trade name of Excel Plumbing Products Ltd.
      </p>
      <hr style="color:red"/>
      <p class="bftr">
      Reg No. 107439 V.A.T. No IE 4800980J Bankers: Bank of Ireland,
        North Circular Rd, Phibsboro, Dublin 7. Bank of Scotland, Ireland, Harcourt Street,
        Dublin 2.</p>
      <hr style="color:red"/>
    </body>
  </html>
</xsl:template>
<xsl:template match="/biztalk:biztalk_1/biztalk:body/basda:Invoice">
  <table width="100%" align="center">
    <xsl:choose>
      <xsl:when test="$swapaddresses = 'false'">
        <tr>
          <td width="50%" align="left">
            <p class="std">
              <xsl:if test="$creditnote='false'">Invoice To:</xsl:if>
              <xsl:if test="$creditnote='true'">Credit To:  </xsl:if>
            </p>
          </td>
          <td width="50%" align="left">
            <xsl:if test="$creditnote='false'">
              <p class="std">Delivered To:</p>
            </xsl:if>
          </td>
        </tr>
        <tr>
          <td width="50%" align="left">
            <xsl:value-of select="/biztalk:biztalk_1/biztalk:body/basda:Invoice/basda:
          /basda:Party"/>
          </td>
          <td width="50%" align="left">
            <br/>
          </td>
        </tr>
        <tr>
          <td width="50%" align="left" valign="top">
            <xsl:apply-templates select="basda:Buyer/basda:Address"/>
          </td>
          <td width="50%" align="left" valign="top">
            <xsl:if test="$creditnote='false'">
              <xsl:apply-templates select="basda:Delivery/basda:DeliverTo/basda:Address"/>
            </xsl:if>
          </td>
        </tr>
      </xsl:when>
      <xsl:otherwise>
        <tr>
          <td width="50%" align="left">
            <p class="std">
              <xsl:if test="$creditnote='false'">Delivered To:</xsl:if>
              <xsl:if test="$creditnote='true'">To: </xsl:if>
          </p>
        </td>
        <td width="50%" align="left">
          <p class="std">
            <xsl:if test="$creditnote='false'">Invoice To:</xsl:if>
            <xsl:if test="$creditnote='true'"> <br/></xsl:if>
          </p>
        </td>
      </tr>
      <tr>
        <td width="50%" align="left">
            <xsl:if test="$creditnote='false'"><br/></xsl:if>
          <xsl:if test="$creditnote='true'">
            <xsl:value-of select="basda:Buyer/basda:Party"/></xsl:if>
        </td>
        <td width="50%" align="left">
          <xsl:if test="$creditnote='false'">
            <xsl:value-of select="basda:Buyer/basda:Party"/>
          </xsl:if>
        </td>
      </tr>
      <tr>
        <td width="50%" align="left" valign="top">
          <xsl:if test="$creditnote='false'">
            <xsl:apply-templates select="basda:Delivery/basda:DeliverTo/basda:Address"/>
          </xsl:if>
          <xsl:if test="$creditnote='true'">
            <xsl:apply-templates select="basda:Buyer/basda:Address"/>
          </xsl:if>
        </td>
        <td width="50%" align="left" valign="top">
          <xsl:if test="$creditnote='false'">
            <xsl:apply-templates select="basda:Buyer/basda:Address"/>
          </xsl:if>
        </td>
      </tr>
    </xsl:otherwise>
  </xsl:choose>
</table>
<br/>
<br/>
<!-- Invoice Header -->
<table width="100%" border="1">
  <tr>
    <td width="17%" align="center" bgcolor="#CCCCCC">
      <p class="hdr"> </p>
    </td>
    <td width="17%" align="center" bgcolor="#CCCCCC">
      <p class="hdr">
        <xsl:if test="$creditnote='false'">Invoice No.</xsl:if>
        <xsl:if test="$creditnote='true'">Credit Note No.</xsl:if>
      </p>
    </td>
    <td width="16%" align="center" bgcolor="#CCCCCC">
      <p class="hdr">A/C No.  </p>
    </td>
    <td width="16%" align="center" bgcolor="#CCCCCC">
      <p class="hdr">Your Ref.</p>
    </td>
    <td width="16%" align="center" bgcolor="#CCCCCC">
      <p class="hdr">
        <xsl:if test="$creditnote='false'">Sales Order Ref.</xsl:if>
        <xsl:if test="$creditnote='true'">No.</xsl:if>
      </p>
    </td>
    <td width="16%" align="center" bgcolor="#CCCCCC">
      <p class="hdr"> </p>
    </td>
  </tr>
  <tr>
    <td width="17%" align="center">
      <p class="std">
        <xsl:apply-templates select="basda:InvoiceDate"/>
      </p>
    </td>
    <td width="17%" align="center">
      <p class="std">
        <xsl:value-of select="basda:InvoiceReferences/basda:SuppliersInvoiceNumber"/>
      </p>
    </td>
    <td width="16%" align="center">
      <p class="std">
        <xsl:choose>
          <xsl:when test="basda:Buyer/basda:BuyerReferences/basda:SuppliersCode
                             ForBuyer">
            <xsl:value-of select="basda:Buyer/basda:BuyerReferences/basda:"/>
          </xsl:when>
          <xsl:otherwise>&#160;</xsl:otherwise>
        </xsl:choose>
      </p>
    </td>
    <td width="16%" align="center">
      <p class="std">
        <xsl:value-of select="basda:InvoiceReferences/basda:BuyersOrderNumber"/>
      </p>
    </td>
    <td width="16%" align="center">
      <p class="std">
        <xsl:value-of select="basda:InvoiceReferences/exchequer:YourRef"/>
      </p>
    </td>
    <td width="16%" align="center">
      <p class="std"></p>
    </td>
  </tr>
</table>
<br/>
<!-- Invoice Rows -->
<table width="100%" border="0" cellspacing="1" cellpadding="2">
  <xsl:choose>
    <xsl:when test="$usebarcode = 'true'">
      <tr>
```

```
        <td width="10%" bgcolor="#CCCCCC" align="center">
          <p class="std">Code</p>
        </td>
        <td width="40%" bgcolor="#CCCCCC" align="left">
          <p class="std">Description</p>
        </td>
        <td width="12%" bgcolor="#CCCCCC" align="left">
          <p class="std">Bar Code</p>
        </td>
        <td width="13%" align="right" bgcolor="#CCCCCC">
          <p class="std">Quantity  </p>
        </td>
        <td width="10%" align="right" bgcolor="#CCCCCC">
          <p class="std">Unit Price</p>
        </td>
        <td width="15%" align="right" bgcolor="#CCCCCC">
          <p class="std">Total</p>
        </td>
      </tr>
      <xsl:apply-templates select="basda:InvoiceLine"/>
    </xsl:when>
    <xsl:otherwise>
      <tr>
        <td width="10%" bgcolor="#CCCCCC" align="center">
          <p class="std">Code</p>
        </td>
        <td width="40%" bgcolor="#CCCCCC" align="left">
          <p class="std">Description</p>
        </td>
        <td width="10%" align="right" bgcolor="#CCCCCC">
          <p class="std">Quantity</p>
        </td>
        <td width="15%" align="right" bgcolor="#CCCCCC">
          <p class="std">Price</p>
        </td>
        <td width="10%" align="right" bgcolor="#CCCCCC">
          <p class="std">Discount</p>
        </td>
        <td width="15%" align="right" bgcolor="#CCCCCC">
          <p class="std">Total</p>
        </td>
      </tr>
      <xsl:apply-templates select="basda:InvoiceLine"/>
    </xsl:otherwise>
  </xsl:choose>
</table>
<br/>
<!-- Totals -->
<table width="100%" border="0" cellspacing="1" cellpadding="2">
  <tr>
    <td width="70%">
      <table border="0" width="100%" height="33">
        <tr>
          <td/>
        </tr>
      </table>
```

```
</td>
<td width="30%">
  <table width="100%" border="0" cellspacing="1" cellpadding="2">
    <tr>
      <td width="15%" align="right" bgcolor="#CCCCCC">
        <font face="Arial,sans-serif" size="1">
          <b/>
        </font>
      </td>
      <td align="right" width="15%">
        <p class="curr">
          <!---
              <xsl:value-of select="translate(//basda:TaxSubTotal/basda:TaxCurrency
              /basda:Currency,'abcdefghijklmnopqrstuvwxyz','ABCDEFGHIJKLMNOPQRST
              UVWXYZ')"/>
          -->
          <xsl:value-of select="//basda:TaxSubTotal/basda:TaxCurrency/basda:
            Currency/@Code"/>
        </p>
      </td>
    </tr>
    <tr>
      <td width="15%" align="center" bgcolor="#CCCCCC">
        <p class="btot">Total </p>
      </td>
      <td align="right" width="15%">
        <p class="tot">
              <xsl:for-each select="basda:InvoiceTotal">
            <xsl:value-of select="format-number(basda:LineValueTotal,'#,##0.00')"/>
          </xsl:for-each>
          <xsl:if test="$creditnote = 'true'"></xsl:if>
        </p>
      </td>
    </tr>
    <!--Omit Settlement Discount if value is zero -->
    <xsl:if test="basda:InvoiceTotal/basda:SettlementDiscountTotal!=0">
      <tr>
        <td width="15%" align="center" bgcolor="#CCCCCC">
          <p class="btot">Less Sett. Disc</p>
        </td>
        <td align="right" width="15%">
          <p class="tot"><xsl:value-of select="format-number(basda:InvoiceTotal/basda:
          ,'#,##0.00')"/>
            <xsl:if test="$creditnote = 'true'"></xsl:if>
          </p>
        </td>
      </tr>
    </xsl:if>
    <tr>
      <td width="15%" align="center" bgcolor="#CCCCCC">
        <p class="btot">VAT Content</p>
      </td>
      <td align="right" width="15%">
        <p class="tot"><xsl:for-each select="basda:InvoiceTotal">
            <xsl:value-of select="format-number(basda:TaxTotal,'#,##0.00')"/>
                  </xsl:for-each>
```

```
                <xsl:if test="$creditnote = 'true'"></xsl:if>
              </p>
            </td>
          </tr>
          <tr>
            <td width="15%" align="center" bgcolor="#CCCCCC">
              <p class="btot">Total</p>
            </td>
            <td align="right" width="15%">
              <p class="tot"><xsl:for-each select="basda:InvoiceTotal">
                  <xsl:value-of select="format-number(basda:NetPaymentTotal,'#,##0.00')"/>
                                </xsl:for-each>
                <xsl:if test="$creditnote = 'true'"></xsl:if>
              </p>
            </td>
          </tr>
        </table>
      </td>
    </tr>
  </table>
</xsl:template>
<xsl:template match="basda:Address">
  <xsl:for-each select="basda:AddressLine">
    <xsl:value-of select="."/>
    <xsl:if test="position() != last()"><br/></xsl:if>
  </xsl:for-each>
</xsl:template>
<xsl:template match="basda:InvoiceLine">
  <xsl:choose>
    <xsl:when test="$usebarcode = 'true'">
      <tr>
        <td align="center">
          <p class="code">
            <xsl:value-of select="basda:Product/basda:SuppliersProductCode"/>
            <xsl:if test="$creditnote = 'true'">
              <xsl:value-of select="basda:Product/basda:BuyersProductCode"/>
            </xsl:if>
          </p>
        </td>
        <td align="left">
          <p class="std"><xsl:value-of select="basda:Product/basda:Description"/></p>
        </td>
        <td align="left">
          <p class="std"><xsl:value-of
          select="basda:InvoiceLineReferences/exchequer:LineUserDef1"/>
          </p>
        </td>
        <td align="right">
          <p class="curr">  <xsl:value-of select="format-
          number(basda:Quantity/basda:Amount,'#,###')"/>
          </p>
        </td>
        <td align="right">
          <p class="curr">
            <xsl:if test="basda:Price/basda:UnitPrice != 0.00">
              <xsl:value-of select="format-number(basda:Price/basda:UnitPrice
                * ((100 - basda:PercentDiscount/basda:Percentage) div 100),'#,##0.00')"/>
                </xsl:if>
            </p>
          </td>
          <td align="right">
            <p class="curr">  <xsl:if test="basda:LineTotal != 0.00">
                <xsl:value-of select="format-number(basda:LineTotal,'#,##0.00')"/>
                          </xsl:if>
            </p>
          </td>
        </tr>
      </xsl:when>
      <xsl:otherwise>
        <tr>
          <td align="center">
            <p class="code"><xsl:value-of
select="basda:Product/basda:SuppliersProductCode"/>
                          <xsl:if test="$creditnote = 'true'">
                          <xsl:value-of select="basda:Product/basda:BuyersProductCode"/>
                          </xsl:if>
            </p>
          </td>
          <td align="left">
            <p class="std"><xsl:value-of select="basda:Product/basda:Description"/></p>
          </td>
          <td align="right">
            <p class="curr">  <xsl:value-of select="format-
number(basda:Quantity/basda:Amount,'#')"/>
            </p>
          </td>
          <td align="right">
            <p class="curr">
              <!-- Need to skip this if unitprice is 0.00 -->
              <xsl:if test="basda:Price/basda:UnitPrice != 0.00">
                <xsl:value-of select="format-number(basda:Price/basda:UnitPrice'#,##0.00')"/>
              </xsl:if>
            </p>
          </td>
          <td align="right">
            <p class="curr">
              <!-- Need to skip this if unitprice is 0.00 -->
              <xsl:if test="basda:Price/basda:UnitPrice != 0.00">
                <xsl:value-of select="format-
number(basda:PercentDiscount/basda:Percentage,'#,##0.00')"/>
                <xsl:if test="string-length(basda:PercentDiscount/basda:Percentage)!=0">
                  <xsl:text>%</xsl:text>
                </xsl:if>
              </xsl:if>
            </p>
          </td>
          <td align="right">
            <p class="curr">
              <xsl:if test="basda:LineTotal != 0.00"><xsl:value-of select="format-
number(basda:LineTotal,'#,##0.00')"/>
              </xsl:if>
            </p>
          </td>
        </tr>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>
  <xsl:template match="basda:InvoiceDate">
    <xsl:value-of select="substring(.,9,2)"/>
    <xsl:text>/</xsl:text>
    <xsl:value-of select="substring(.,6,2)"/>
    <xsl:text>/</xsl:text>
    <xsl:value-of select="substring(.,1,4)"/>
  </xsl:template>
</xsl:stylesheet>
```

Appendix 5

List of Figures

List of Figures (cont'd)

Appendix 6

List of Tables

Bibliography

Publications

Avison, D.E. & Fitzgerald, G. (1995).	*Information Systems Development: Methodologies Techniques & Tools* McGraw Hill
Castro, E. (2001).	*XML for the World Wide Web.* Peachpit Press
Forouzan, B. A.	*Data Communications and Networking* McGraw Hill
Harold, R.E. (2001).	*XML Bible* Hungry Minds
Pressman, R.S. (1997).	*Software Engineering: A Practitioner's Approach* The Process. McGraw Hill. 22 – 48 (Chap. 2)
Sommerville, I. (2001).	*Software Engineering* Software processes, process models. Addison Wesley. 44 – 55 (Chap. 3)
Tittel, Ed. (2004).	*XML Crash Course.* (Chaps. 3, 6, 7) Schaum's Outlines McGraw Hill

Web Sites

Topic	Address
ADOS	http://www.canon.co.uk/for_work/products/document_imaging_systems
BASDA	http://www.basda.org
e-BIS XML	http://xml.coverpages.org/xmlnews2000Q4.html http://xml.coverpages.org/ http://xml.coverpages.org/ebis-xml.html http://xml.coverpages.org/eBIS-XMLschema_invoice_v301-xml.txt
Exchequer	http://www.exchequer.com
Scanfile	http://www.canon.co.uk
XML	http://www.w3.org/XML/#intro http://www.w3.org/TR/xslt http://www.w3.org/TR/xpath