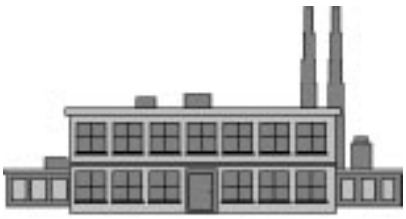
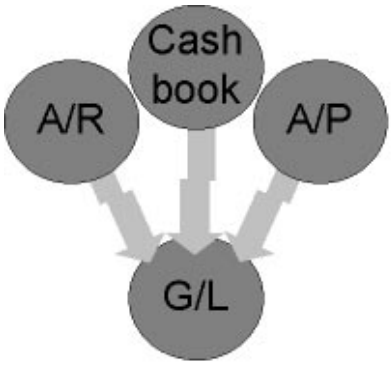
SYNON/FINANCIALS
INTEGRATED DATA WAREHOUSING



WHAT IS DATA WAREHOUSING

Simply defined, a *Data warehouse* is populated by *informational data* created by denormalising and summarising *operational data* so that it is appropriately formatted for analysing the business.



The modular approach is fundamentally flawed

DATA WAREHOUSING AND FINANCIAL SYSTEMS

With regard to financial accounting and management systems, *data warehousing* has quickly achieved prominence because of the fundamental weaknesses inherent in systems design. The modular approach that has always been taken with stand alone G/L, A/P, A/R and Cash management applications involves passing data between them, either in batch or real-time. This makes simple *transformation* of the *operational data* into *informational data* impossible.

For example:

- Data held in the G/L is typically the summarised G/L impact of groups of transactions, such as A/P vouchers, held in subordinate ledgers. Ascertaining the origin of G/L postings is therefore difficult, relying on tenuous audit trails which are easily corrupted and often paper based.
- Accessing data in the G/L is traditionally achieved using an account number with an imbedded hierarchy. This hierarchical code does not relate directly to real world business objects, and does not reflect the true dynamics of the corporate environment. Detailed multi-dimensional analysis of the wealth of data locked away in the G/L is therefore very difficult. Coping with organisational changes is impossible without mass recoding and the attendant impact on fragile audit trails.
- Complex trading relationships are difficult to monitor and control. A customer in the A/R ledger may also be a vendor in the A/P ledger. With stand alone ledgers, maintaining a view of total corporate exposure is extremely difficult. This is especially true when the customer and vendor codes are totally different.
- Data is often held in incompatable formats in different applications. Codes and description may be of different length. Value fields may have different formats. Even data relating to the same business entity may be different. For example a vendor 'Trading Partner Inc' in the A/P ledger may be described as 'T.P.I.' in the A/R.

As a result there are now many tools on the market that will perform the data *cleansing, transformation* and *propagation* required to take *operational data* (from both relational and non-relational systems) and turn it into *informational data* that can easily be analysed, usually in an entirely separate *Data warehouse*. This means you cannot drill down on-line to the *operational data* to verify the accuracy and source of the *informational data,* which is a major drawback.

The cost of these tools is significant and their implementation is highly complex. However, the benefits from effective implementation of a *Data warehouse* justify the investment.

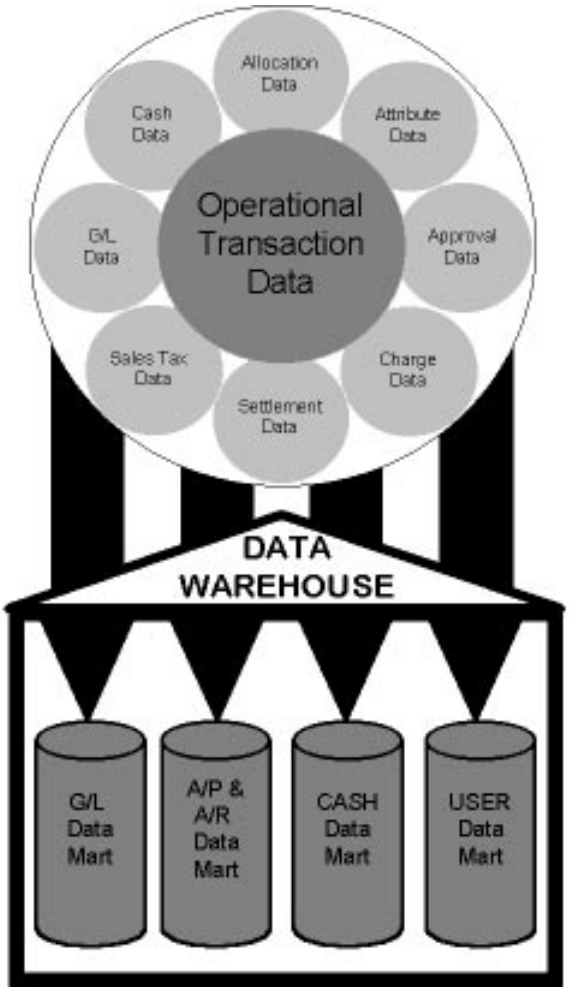
New Technology based on current availability

Although *Data warehousing* is considered a new technology, it has been possible for several years to build systems that combine the *OLTP* performance required to handle *operational data* and the *OLAP* functionality to enable multi-dimensional analysis of that data.

Such systems must be based on fully integrated relational database technology, where real business objects can be defined as database entities inheriting attributes from higher order objects. The design needs to incorporate denormalised summary files based on the higher order objects so that information about those objects can be quickly and easily analysed. They should be maintained in real time to ensure the currency of the information, with minimal impact on *OLTP* performance.

Synon/Financials Integrated Data Warehouse

Synon/Financials is just such a system. A 'clean sheet' development of the 1990s, it carries none of the baggage of either legacy- or process- oriented systems. It is a complete suite of financial management applications fully integrated at database level. Its model-based design provides a very open object-oriented database architecture. *Operational data* is held in files normalised to maximise *OLTP* performance. The *informational data, propagated* in real-time is denormalised to maximise speed of access and allow multi-dimensional analysis. A powerful integrated *OLAP* engine is provided over the G/L *data mart* within the *Data warehouse* to unlock the wealth of information held in the G/L.



FULLY INTEGRATED ARCHITECTURE

Fully integrating *operational and informational data* in the same application is only possible because of Synon/Financials advanced design.

Unlike traditional systems there are no batch processes which update ledgers by moving and summarising data from one file to another. All the *operational data* is held in a set of closely related database files from which it never moves. Robust audit trails are maintained and duplication is minimised.

Posting a financial transaction to the books is therefore merely a case of changing the status of the relevant related records. This means that all the ledgers are updated simultaneously with the *informational data* being *propagated* to the relevant *data marts* in real-time. The simplicity of this process means that *OLTP* performance is maintained and the accuracy of the data stored in the *Data warehouse* guaranteed.

The processing makes full use of commit control functionality so that if the transaction processing is interrupted for any reason full rollback is achieved.

Ledgers can never be out of balance and the *operational* and *informational data* is always fully synchronised.

Arguably one of the most important commercial benefits of such complete integration is the full drill down and roll up capability that results. From any *data mart* you can drill down to the *operational data,* including the audit stamps which are held on every record, thereby verifying the quality and accuracy of the management information provided by the system.

OBJECT ORIENTATED DATABASE DESIGN

Data transformation and *propagation* is simplified in Synon/Financials because of the object-oriented approach taken in the design of the database. Effective use of inheritance provides real benefits for the user. It minimises the maintenance requirement and maximises operational productivity.

Business objects defined

Synon/Financials allows for the user definition of real world business objects such as:

Administrative division - analogous to Corporation or holding company.

Financial division - typically a trading company within a Corporation

Ledger - a book of accounts

Personal ledger - a subordinate ledger used to manage trading relationships with other organisations.

Cash book - a subordinate ledger used to manage a bank account or petty cash fund.

Entity - a trading partner. This includes customers, vendors, employees, operating units, companies and, indeed, any business entity that has a name and address. All entities are recorded once and once only in a central repository which is used throughout the system.

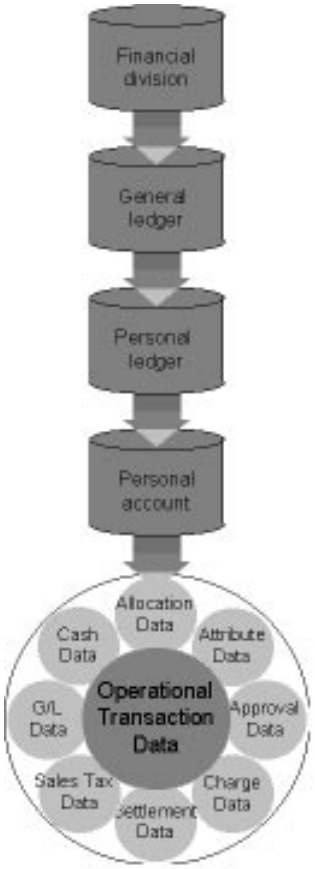
Personal account - a business object which associates a Personal ledger with an Entity thereby defining a specific trading relationship with that Entity. Multiple trading relationships can be easily established for any Entity. Corporate exposure can therefore be viewed via the *data mart* by either Personal ledger or Entity across company and ledger.

Unit - an operating unit within the corporation. A unit could be a profit centre, a cost centre, a sales territory, a department or the corporation itself.

Period - a user defined time frame between two dates identified for accounting purposes. There is no limit to the number of periods that can be included in an accounting year. There is no restriction on their length and periods of varying length can be included in the same financial year.

Transaction - a financial transaction. Transactions are classified and their behaviour specified using other user defined objects such as **Transaction types**, **Transaction templates** and **Transaction numerators**.

Inheritance enhances productivity



Effective use of inheritance

Effective use of inheritance is a significant aid to productivity. It allows common data to cascade down the inheritance hierarchy requiring only exception data to be entered at the lower levels.

For example:

- The General ledger inherits the properties such as operating currency from the Financial division it is used to control.
- The Personal ledger inherits properties such as accounting period and other control information from the General ledger to which it is subordinate.
- The Personal account inherits properties from the Personal ledger and Entity to which it belongs.
- Low level objects such as Transactions inherit the properties of the Personal accounts they are associated with.

Business metadata

The high level business objects such as Financial division and Personal ledger, defined by users, hold the *business metadata* around which the *data marts* in the *Data warehouse* are constructed. As new high level objects are defined, Synon/Financials automatically extends the appropriate *data mart* to cope with the additional *informational data* that will be created.

MODEL BASED OPEN ARCHITECTURE

Synon/Financials functional and database architecture is very open. Both the *operational* and *informational data* is clearly defined, held in a central set of files and thereby easily accessed.

Technical metadata

Synon/Financials is a model based development. The model is the high level description, or *Technical metadata*, of the entire functional and database design.

The Synon/Financials model is made available so it can be used to customise, extend or enhance any

aspect of the system with a great deal more ease than is possible with other systems.

Users have the ability to extend operational functionality, expand existing *data marts*, create additional inquiries, reports or downloads and even introduce new completely user defined *data marts*.

Functional APIs

A complete set of transaction APIs is provided to allow data to be passed in from other systems and subsequently processed by Synon/Financials to update the ledgers and the *Data warehouse*.

In addition both pre- and post-processing APIs have been incorporated in the transaction processing functionality, so that it can easily be extended to update other systems and additional *data marts* which can be created.

These APIs are all fully documented in the Development Guide, a document provided specifically to assist developers customise, enhance and interface to Synon/Financials.

User defined data

Synon/Financials incorporates the concept of user definable data. The user can define additional data items, known as "attributes", as required and with no re-programming. Simple inquiries based on select/omit processing are provided over attributes. They can also be used by the G/L *data mart OLAP* engine to further subset data extraction.

Their real power becomes apparent when they are used as the basis for user defined *data marts* which can be constructed using the *technical metadata* in the application model.

Business intelligence software tool of choice

Synon/Financials provides pre-defined inquiries and reports over the *data marts* in the *Data warehouse* to allow easy access to the *informational data* stored within them. In addition a powerful *OLAP* engine is provided for the G/L *data mart* to enable users to slice and dice the data any way they wish and drill down to and roll up from the *operational data*.

However as *Data warehousing* matures as a technology, more and more *Business intelligence software tools* are becoming available. These tools are becoming more sophisticated, with the latest *data mining* tools using powerful algorithms to search out patterns, classifications and associations not immediately obvious.

Many financial accounting software providers are forming alliances with these tool providers in an attempt to overcome the inherent flaws in the underlying design of their software.

Synon/Financials adopts the completely opposite approach, allowing absolute freedom of choice to its users. The open database architecture with all the *operational data* held in closely associated database files, means that users can adopt the Business intelligence software tool of their choice. They are not restricted by the software vendor as a result of other strategic or tactical agendas the vendor many have.

SYNON/FINANCIALS' DATA WAREHOUSE

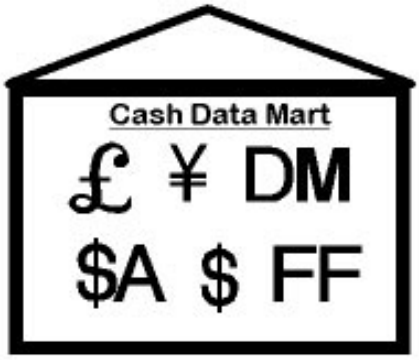
The *Data warehouse* within the Synon/Financials product is comprised of three *data marts*. They are the:

- General ledger *data mart*
- Personal ledger *data mart*
- Cash book *data mart*

These *data marts* vary in complexity. They range from the Cash book *data mart* at the simplest level to the highly sophisticated General ledger *data mart* with its powerful, fully integrated *OLAP* engine.

Complete functionality is provided over these *data marts* to view the information they contain. Using the *technical metadata* the *data marts* and the functionality provided can be extended. Additional information can be stored and additional inquires, reports or downloads built. Completely new *data marts* can be constructed.

Cash book Data mart



The Cash book *data mart* is the simplest of the three marts provided.

For each bank account or petty cash fund identified by the user, Synon/Financials automatically maintains both actual and reconciled cash balances in the currency of the Cash book.

These balances for every Cash book are all displayed on a single panel. An up to date view of your cash position is therefore always available.

From this single display, inquiries are provided to:

- Display the Cash book definition, including Cash book type, control accounts, bank details, currency, usual settlement method, and so on.
- Drill down to all the cash items recorded in the Cash book to the lowest level of detail.
- Drill down to the complete transactions from which the cash items originated.

Where cash has been applied to either payable or receivable invoices, the invoice numbers are available via either of the above drill downs.

Personal ledger Data mart

More sophisticated than the Cash book *data mart*, the Personal ledger *data mart* holds a wealth of information about your business relationships with all your trading partners.

Currency translation is performed on-line, so values can be displayed in any currency.

The *informational data* can be viewed by:

- Entity (Trading partner) across all companies and ledgers
- Ledger for all Entities

The following information about your relationship with a trading partner, across both companies and ledgers, can be viewed from a single point:

- Total net balance outstanding
- Total cash received but not yet applied
- Total value of transactions raised
- The last transaction raised - payable or receivable
- The last cash transactions
- Average number of days taken to settle open items
- Average number of days early or late settling open items

When a company and ledger are supplied the following additional information is immediately made available:

- Account balance outstanding
- Total cash unmatched or overpaid
- Credit limit
- Account status
- Account Contact
- Usual method of settlement
- Open item values summarised as :
  - Unapproved
  - In dispute
  - Approved
  - In total

From either of these summary panels a single keystroke gives access to more levels of detail on specific areas of interest, all the way down to the underlying *operational data* from which it was *propagated*.

The following is a summary of the *informational data* available:

- Cash payments or receipts which have not yet been applied to open items.
- Cash applied to invoices that have not yet been received or have been overpaid.
- Trading statistics by accounting period showing
    - ~ Total value of invoices processed
    - ~ Total value settled
    - ~ Number of transactions processed
    - ~ Average days to settle open items

Further drill down within an accounting period provides, for either all transactions or by transaction type, the following information:

- ~ Total value of invoices processed
- ~ Total value settled
- ~ Average days to settle
- ~ Average early or late
- ~ Settlement discount taken
- ~ Settlement discount not taken
- ~ Value of Gains/losses on currency exchange
- ~ Value of open item corrections
- ~ Value of write offs
- ~ Value of all debit notes raised
- ~ Value of overpayments
- ~ Detailed settlement discount statistics

- All personal account transactions
- Open items listing selections
- All transactions
- All cash items
- Open items
- Aged balances and open items within ageing periods

General ledger Data mart

The General ledger *data mart* is the most sophisticated of the three *data marts* provided, incorporating a fully integrated Executive Analysis System (EASy).

This powerful *OLAP* engine employs user defined *business metadata* to allow detailed analysis of the wealth of data traditionally locked away in the General ledger.

Answers to business questions can quickly be determined by slicing and dicing through the data, or drilling down or rolling up different hierarchical levels of the data.

The user defined *business metadata* consists of totalling and sequencing structures, value formatters, user defined algorithms and inquiry/report formatters.

There is no limit to the number of these items of *business metadata* that can be defined, thereby providing virtually limitless analysis of available information.

The totalling and sequencing structures are user defined hierarchies of the business objects recorded against the values in the General ledger. These business objects define:

Who - the operating *unit* to which the posting belongs

What - the *nature* of the account to which the posting was made.

When - the *period* in which the posting has to be accounted for.

Combining these hierarchies of business objects creates multi-dimensional data structures which can be used to answer questions such as:

from a sales manager -

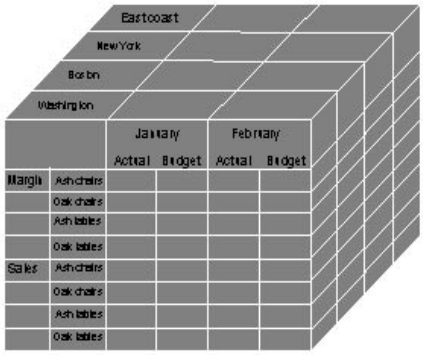
- √ What is the likelihood of achieving next quarter's targets?
- √ Which sales region is the most profitable?.

from a financial controller -

- √ Is there a correlation between doubtful debt and sales turnover?

from a wholesaler -

- √ What are sales by product by region over time?
- √ How do the figures compare against forecast?



Multi-dimensional data structures

from a merchandise manager -

- √ What is the profitability by product line and does it vary over the year?

from a furniture manufacturer-

- √ Is there a movement from ash products to teak products?

from an insurance company -

- √ What is the variance between claims and premiums by type of risk insured?
- √ Is it different by region?

from a retailer -

- √ What are the trends in summer sales turnover for the last five years?

from a chief executive-

- √ What would be the effect if we consolidated these units?
- √ How robust is sales growth in the Eastern region versus the rest of the nation?

The Value formatter and algorithms in the *business metadata* allow the business analyst to define what and how values are presented. The algorithms provide the means by which values held in the *Data warehouse* are manipulated to show, for example:

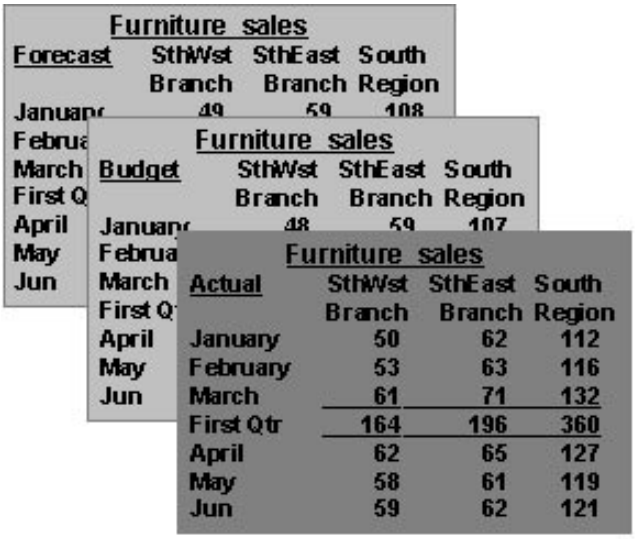
- Actual less budget
- Actual to forecast variance
- Sales revenue divided by sales volume to give average unit sales value
- Costs as a percentage of total sales
- This year against last year

As these algorithms are all user defined, they are limited only by the analyst's imagination.

The report/inquiry formatter determines how the *multi-dimensional data structures* are presented. This means that the same data structures can be used to answer fundamentally different questions.

For example, the two analyses below show different presentations of the same *multi-dimensional data structures*. The first answers the question "What are sales by branch over time?". The second addresses the question "How are the branches performing against budget by product?"

Sales by branch over time



Furniture sales

Forecast	SthWst Branch	SthEast Branch	South Region
January	49	59	108
Februa			
March			
First Q			
April			
May			
Jun			

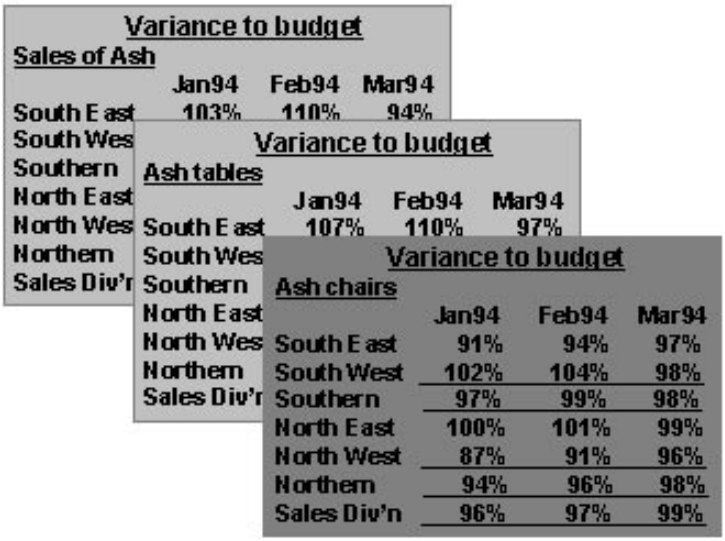
Furniture sales

Budget	SthWst Branch	SthEast Branch	South Region
January	48	59	107
Februa			
March			
First Q			
April			
May			
Jun			

Furniture sales

Actual	SthWst Branch	SthEast Branch	South Region
January	50	62	112
February	53	63	116
March	61	71	132
First Qtr	164	196	360
April	62	65	127
May	58	61	119
Jun	59	62	121

Variance to budget



Variance to budget

Sales of Ash

	Jan94	Feb94	Mar94
South East	103%	110%	94%
South Wes			
Southern			
North East			
North Wes			
Northern			
Sales Div'r			

Variance to budget

Ash tables

	Jan94	Feb94	Mar94
South East	107%	110%	97%
South Wes			
Southern			
North East			
North Wes			
Northern			
Sales Div'r			

Variance to budget

Ash chairs

	Jan94	Feb94	Mar94
South East	91%	94%	97%
South West	102%	104%	98%
Southern	97%	99%	98%
North East	100%	101%	99%
North West	87%	91%	96%
Northern	94%	96%	98%
Sales Div'n	96%	97%	99%

Analyses created using EASy functionality can be either

- Output as a report which can be either printed or viewed interactively.
- Displayed on-line with full drill down and roll up capability through the various levels and dimensions of the data structures and back to the *operational data*.

- Downloaded in PC format for import into any PC based analysis tool for further manipulation, graphical presentation or inclusion in a text based report.

User defined Data marts

As well as extending the *data marts* provided, Synon/Financials' open architecture dramatically simplifies the process of creating additional bespoke *data marts*.

Technical metadata

The *technical metadata* is the high level business model that describes the entire functional and database design. From this model the complete application can be automatically regenerated.

Providing the model with the application enables the high level description and subsequent generation of additional *data marts*, further enhancing the significant benefits that flow from Synon/Financials' *Data warehouse*.

Transaction processing API's

Both pre- and post- processing APIs are provided as part of the transaction processing suite of functions.

Because of Synon/Financials advanced object orientated database design, it has been possible to incorporate all the data *cleansing* and *propagation* functionality, required to populate the *Data warehouse,* in the transaction processing functionality without detrimentally impacting *OLTP* performance.

To populate bespoke *data marts* the transaction processing functionality needs to be extended. This can be achieved very effectively without the risk of jeopardising delivered functionality by using the transaction processing APIs.

Soft attributes

Synon/Financials supports the concept of user defined (or soft) attributes. These allows users to define whole new classes of data to be captured without the need for any re-programming.

Bespoke *data marts* can therefore be populated either by the *operational data* held in the main transaction files or by completely user defined *operational data* specified using soft attributes.

No matter what your business requirements are in terms of financial information Synon/Financials provides you with the power to address them.

CONCLUSION

The benefits that are available from *data warehousing* are significant. They accrue from the ability to slice and dice through the data, or drill around different hierarchical levels of *multi-dimensional data*, thereby delivering the management information needed to address real business issues, rather than displaying simple historic data.

The cost of implementing data warehousing with traditional systems is often greater than the cost of the operational application.

Synon/Financials unique database architecture enables the incorporation of these benefits as an integral part of the product at no additional cost.