

Exchequer Enterprise v4.32

Interface Briefing

Introduction

“ No accounting system can afford to be an island divorced from a business's other information systems” – Eduardo Loigorri, Managing Director, Exchequer Software Ltd.

This statement forms the cornerstone behind the philosophy Exchequer Software have adopted in allowing data to flow both to and from Exchequer Enterprise.

Many approaches have been taken to allow both data extraction and data insertion from sources external to Enterprise's core accounting modules. Since no single technology can provide all the data manipulation capabilities required, several different mechanisms have been used to provide as much access as possible. The common thread between them is the Windows environment.

This document seeks to explain the various technologies used by Exchequer Software, and illustrate their implementation. In some cases there is an overlap between technologies that may provide the same data access by different means.

Technology Overview

COM Toolkit

The COM Toolkit is a next generation toolkit for linking applications with Enterprise in a future proof language independent way.

Customisation

There are often instances when additional features are required in order to satisfy a particular users requirements, and it is rarely desirable for the core system to be modified for just a single user. To this end an event handler has been incorporated within Enterprise to allow external routines to take control of some parts of Enterprise's program flow.

The first event implemented was to allow our overseas distributors to customise the Enterprise VAT calculation for local requirements. Since then many more events have been implemented in Enterprise providing increased functionality and flexibility.

Although Enterprise's main screens are not alterable, it is possible to add new sections or popup boxes with additional fields as required. Several examples are available for dealers to examine, including an Uplift Hook and a User-Defined Field Hook.

When Enterprise encounters an event, it calls the Enterprise event handler passing a suite of objects that define the event and its data, and then waits for the handler to return after completion.

From Enterprise v4.31 the Event Handler can either be built into a DLL written in Delphi 5.0, or in an Out-Of-Process COM Client written in any language supporting COM. Multiple Event Handlers can be linked to Enterprise using any mixture of the two methods.

Import Module

The Import Module provides a batch method of importing data into Enterprise in MS-DOS and 32-Bit Windows. Its functionality does overlap with that of the Toolkit DLL, which can be used to provide the same service in real-time. The import module is covered in more detail in a separate document.

ODBC

The Enterprise ODBC Kit provides a read-only interface to the raw data inside Enterprise. This is usually used for reporting purposes in conjunction with a package like Crystal Reports or MS-Access, but can be used in bespoke systems for data extraction.

ENTFUNCS.DLL

This is a utility DLL written for the Enterprise ODBC Kit to give access to the Enterprise General Ledger History figures.

From v4.31 the licencing terms have been extended to allow this DLL to be used by anyone with a valid licence for the ODBC kit or the Toolkit DLL.

What is the 'COM Toolkit'?

The COM Toolkit is a next generation toolkit for linking applications with Enterprise.

By using the COM technology we have made the COM Toolkit accessible to many of the common applications and development tools currently in use, and many that haven't even been released yet, like .NET

Beyond the technical definition, the COM Toolkit opens up a world of opportunity and possibility of extending Enterprise.

The fact that Office applications like Outlook support COM at their core means it is possible to integrate Enterprise to a granular level with Office applications, like presenting a real-time balance in an Outlook form, to developing major applications which integrate with Enterprise.

Combine this with the COM Customisation and you have a formidable array of Enterprise customisation tools.

What Advantages does it have?

So what does the COM Toolkit give you that other tools like the Toolkit DLL or Import Module don't?

Ease of Use

The COM Toolkit has been designed from day 1 to be as easy as possible to use, whilst not limiting its power.

It presents developers with a more familiar object-oriented interface, completely different to the Toolkit DLL's API interface that many modern programmers find intimidating.

Also wherever possible the business rules have been automated behind the scenes, lowering the level of Enterprise knowledge that programmers need in order to link to Enterprise.

Language Independence

The COM Toolkit has already been successfully used with more languages and tools than the Toolkit DLL, the following languages or tools have been successfully used:-

- Borland Delphi
- Microsoft Access
- Microsoft FoxPro
- Microsoft Visual Basic
- Powerbuilder
- Microsoft Visual Basic.NET (*Alpha release*)

Additionally it has been successfully used with the following applications:-

- Microsoft Excel 95, 97 and 2000
- Microsoft Outlook 2000
- Microsoft Word

Forward Compatibility

The very nature of the COM Toolkit allows us to make changes in future versions of the COM Toolkit without affecting existing programs.

This is dramatically different to Toolkit DLL programs that need to be rebuilt whenever we have to change the structures, or to systems that use the Import Module which periodically need to change the file structure.

Distribution

The COM Toolkit is also going to be installed with Exchequer Enterprise and automatically configured, so you don't have to worry about installing and configuring it.

Also Patch programs will be available on the Exchequer web-site to upgrade the COM Toolkit to the latest version with just a couple of clicks. An example patch program can be found on the event CD.

Flexibility

The nature of COM makes it easier to link to other COM compliant applications very simply. Closer integration between apps is the cornerstone of the Microsoft.NET initiative. The COM Toolkit is Exchequer Software's response to this challenge, to act as the glue between Enterprise and the outside world.

The scale of integration is infinitely variable making the COM Toolkit ideal for minor add-in's to full blown applications.

What Can the Toolkit Do?

The Exchequer Enterprise Toolkit allows you to read from and write to the Enterprise database in a way that doesn't require detailed knowledge of the underlying business rules. It hides those rules from you and automatically processes them as and when required.

The Toolkit can be used with either Exchequer or Exchequer Enterprise data in Windows 9x, Windows NT and Windows 2000. Windows CE and MS-DOS have no equivalent of the Toolkit DLL.

The Toolkit provides functions that allow you to access the following data categories: -

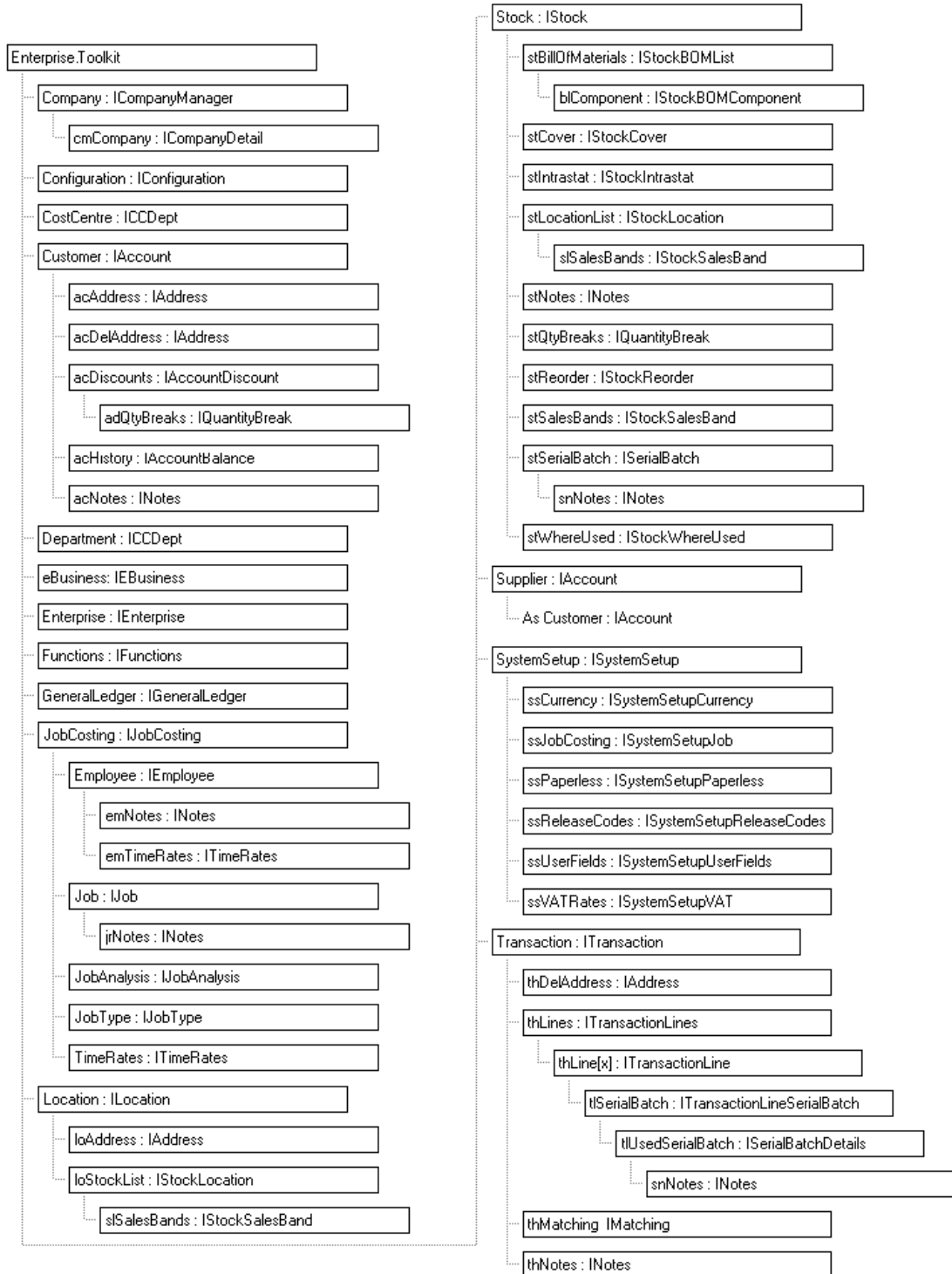
- Customers
- Suppliers
- Transactions, including a limited ability to update existing transactions.
- Stock, including Batch/Serial Numbers, Bill Of Materials and Locations.
- General Ledger Codes
- Job Costing, including Analysis Codes, Employees, Job Types and Time Rates.
- Customer, Supplier, Stock and Transaction Notes
- System Setup, including Currencies and VAT Rates.

Other functions are also available for specific purposes such as converting amounts between currencies.

The toolkit is ideally suited for integration between Enterprise and external systems which require real time seamless links to and from Enterprises' rich databases, with as much forward compatibility with future versions of Enterprise as possible. The COM toolkit delivers that objective.

Object Model

The interface of the COM Toolkit for programmers is called the object model. The Object Model for the v1.00 release looks like this:-



Additionally within a number of the objects are properties that return a related object, for example the Transaction.thAcCode property returns an IAccount object containing details about the Transactions Customer or Supplier.

These property objects have not been shown in the diagram, as that would make it too complex to draw, let alone understand.

Here is a brief overview of each of the top-level objects:-

Enterprise.Toolkit

This is the main object of the COM Toolkit from which all other objects are accessed. It also contains the COM Toolkit Version number, and the OpenToolkit and CloseToolkit methods that are used to open and close the Toolkit.

Company : ICompanyManager

This object lists the companies defined in the Multi-Company Manager. The Multi-Company Manager directory is determined by the Configuration.EnterpriseDirectory property, the Company object is not available if Configuration.EnterpriseDirectory is not correctly set.

Configuration : IConfiguration

This object contains the settings which control how the COM Toolkit works, the initial values are loaded from ExchDll.Ini, but can be changed before the OpenToolkit method is used to open a set of company data.

CostCentre ICCDept

This object provides a read-only interface into the Cost Centre database.

Customer : IAccount

This object provides access to the Customer database, and to details associated with customers, for example customer discounts, customer notes and the customer history.

Department : ICCDept

This object provides a read-only interface into the Department database.

Enterprise : IEnterprise

If Enterprise is running on the same workstation at the same time as the COM Toolkit, this object will provide information on what currency version it is, which company is being used, who is logged in, etc...

NOTE: This feature is only supported in Enterprise v4.32 and later, and if multiple copies of Enterprise are running on the same workstation the data could be a mix of information from both copies.

Functions : IFunctions

The Functions object provides various functions that don't fit in anywhere else in the object model, for example entRound is used to round a double precision number to a specified number of decimal places.

One of the more interesting functions is entBrowseObject, this allows you to pass in any of the COM Toolkit objects, and to see its properties, and those of any sub-object. This is very useful when debugging an application.

GeneralLedger : IGeneralLedger

The General Ledger object provides access to the General Ledger Codes database.

JobCosting : IJobCosting

The JobCosting object encapsulates all the JobCosting functionality in the COM Toolkit.

Location : ILocation

This object provides access into the Locations database, and the loStock sub-object lists the Stock details for the current location.

Stock : IStock

This object provides access into the Stock database, and to details associated with Stock. The stLocations sub-object lists the Locations that the Stock is stored at, and the stNotes sub-object provides access to the Stock Notes.

Supplier : IAccount

This object provides access to the Supplier database, and to details associated with suppliers, for example supplier discounts, supplier notes and the supplier history.

SystemSetup : ISystemSetup

This object provides read-only access to the System Setup information from Enterprise. As this information is mainly static the data is cached for better performance, the Refresh method can be used to reload the cache should your program need it.

Transaction : ITransaction

The Transaction object provides access to the Transactions database, and to the information associated with Transactions.

The thLines sub-object provides access to the Transaction Lines for the current transaction, the thMatching sub-object provides access to the Financial and SOP matching information, the the thNotes sub-object the Transaction Notes.

Introduction To Customisation

What is Customisation?

The Exchequer Enterprise Customisation is an interface that allows external code to be linked into the Enterprise program flow, to alter the standard behaviour of Enterprise.

Alterations made through the Customisation can be as simple as defaulting a value, or as major as replacing the VAT and Period calculations.

Customisation is of particular interest to dealers as it can often enable a sale to be made, where the standard Enterprise functionality isn't sufficient for a customer's particular requirements.

What Can Customisation Do?

The customisation is typically used for the following purposes:-

- Defaulting field values.
- Additional field level or record level validation.
- Improved integration with external systems.
- Enhanced Line Calculations, e.g. popup window for calculating Quantity by Area.
- Automatic calculation and insertion of additional transaction lines for Carriage Charges or Discounts.
- Replacing the VAT Calculation for overseas customers.
- Replacing the Period numbering system for non-standard periods.

What Can't Customisation Do?

The customisation does have limits which developers need to be aware of:-

- Each hook has to be specially coded within Enterprise.
- A hook cannot deal with data outside the scope of the event.

How Can The Customisation Be Extended?

Although very useful, the customisation is limited to dealing with single events, which in turn means dealing usually with a single field.

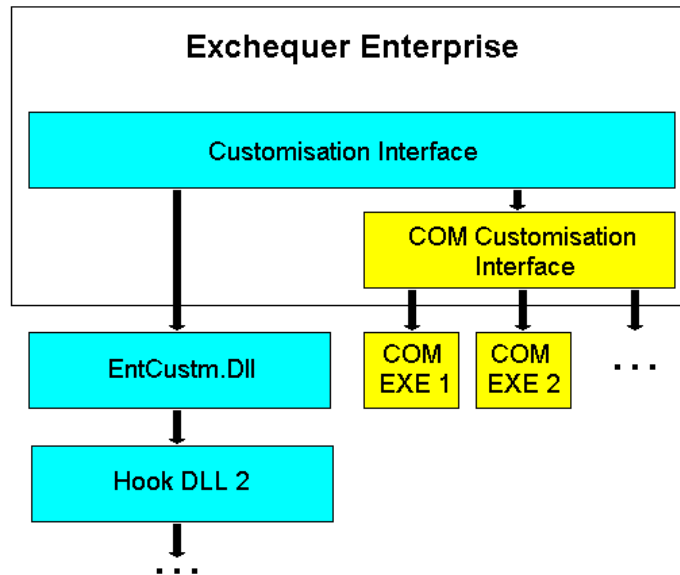
To extend the capabilities of the hooks to cope with larger customisation tasks such as dealing with multiple transactions during a single hook event, it is possible to combine the Customisation with the Toolkit.

A good example of using the Toolkit with Customisation would be the Enterprise Uplift Hook, this uses customisation to integrate an Uplift dialog into Purchase Transactions, and uses the Toolkit for updating the Transaction.

Enterprise Customisation Subsystem

From Enterprise v4.31 there are two customisation methods available to developers:- Delphi 5.0 Customisation and COM Customisation.

The diagram below illustrates how the two customisation methods interact with Enterprise and each other:-



As the diagram shows, the COM Customisation is a client of the Delphi 5.0 Customisation.

After the Delphi 5.0 Customisation has called any of its own clients, it will call the COM Customisation so the COM clients can be called.

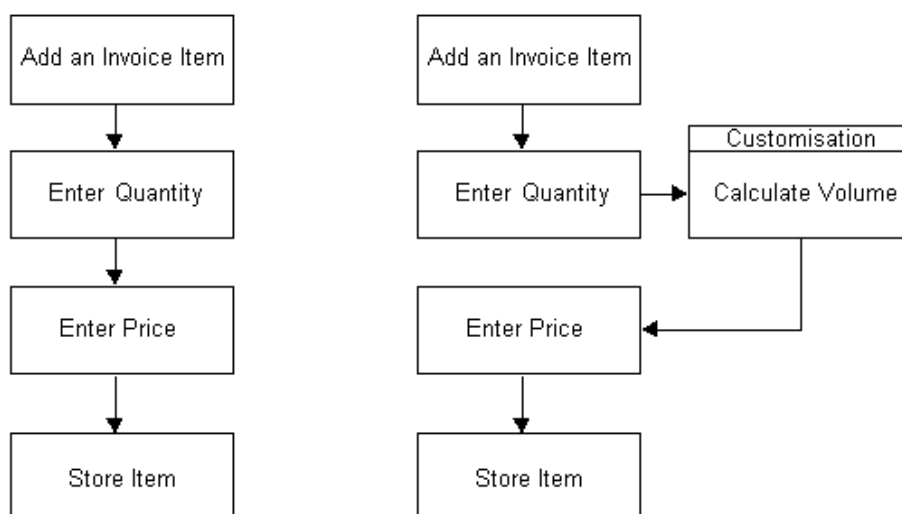
Types Of Customisation

There are three distinct sub-types of customisation:- Hooks, Label and Menu. Hook Customisation is by far the most useful and powerful.

Hooks Delphi 5.0 + COM

This is the most useful type of customisation, Hooks allow the integration of external code into the Enterprise program flow.

This diagram illustrates how customisation can be integrated into the program flow, without changing significantly how Enterprise works:-



The normal program flow is shown on the left, the same program is shown on the right with the addition of a hook on the Quantity field. It should be emphasized that all the normal validation is still done.

Custom Buttons

A number of hooks don't integrate with the existing program flow like that shown above, but create new program flow.

Custom Buttons are hidden buttons in Enterprise that can be enabled using Hooks. When the user clicks the Custom Button, the customisation is called to perform the required operations. This is a nice way to integrate external or toolkit operations into Enterprise.

Custom Buttons also use Label Customisation to set the caption on the button.

Label Delphi 5.0 only

Label Customisation allows the Customisation Client to modify textual descriptions within Enterprise. Currently the modifiable text is limited to the captions on custom buttons.

See the Exercise later in this training course for more detail on Label Customisation.

NOTE: This customisation type is not yet available in COM Customisation, a Delphi 5.0 Customisation DLL has been written to provide Label customisation to COM Customisation developers.

Menu Delphi 5.0 only

Menu Customisation allows the menu structure of the main Enterprise window to be modified, popup menus cannot be modified

The Tools menu in Enterprise is a good example of menu customisation, originally written as a demo for the technology, it was so popular it now ships as standard with Enterprise.

NOTE: This customisation type is not yet available in COM Customisation.

COM Customisation

COM Customisation was added in Enterprise v4.31 to provide a language independent method of customising Enterprise, that was also simpler to for developers to use.

As the name suggests, COM Customisation is COM based, with a COM Customisation Server being implemented within Enterprise. COM was chosen because it gives us the language independence that we wanted.

It is expected that any development tool supporting COM can now be used to customise Enterprise, during development we have used test clients written with Borland Delphi versions 3, 4, and 5, Borland C++ Builder 4, and Visual Basic versions 5 and 6.

What Does A COM Client Do?

This section describes the general interaction between a COM Client and Enterprise, it is intended as a general overview, for language specific details see the Exercises later in this training course.

A COM Client is a .EXE which connects to the Enterprise Customisation Object within Enterprise. First the client should try to connect to an existing instance of the COM Object, if that fails then it should create a new instance.

When the Customisation Object is created it automatically registers itself in the Running Object Table (ROT) in Windows. This allows multiple .EXE's to use the same object, this simplifies the code required in Enterprise and improves performance whilst reducing resource usage.

Once the Client has successfully connected to the Customisation Object it has access to the following object:-

```
ICOMCustomisation
├── SystemSetup : ICOMSetup
│   ├── ssCurrency : Array of ICOMSetupCurrency
│   ├── ssVATRates : Array of ICOMSetupVAT
│   └── ssUserFields : ICOMSetupUserFields
└── VersionInfo : ICOMVersion
```

This can then be used to enable specific hooks within Enterprise. It can also add a textual description into the main Enterprise Help-About dialog, this helps technical support later.

The Client must also respond to two COM Events on the object:- OnHook and OnClose.

The OnHook event executes whenever a Hook is reached in Enterprise that has been enabled by a COM Client. Enterprise cannot distinguish between COM Clients, so each event has to be sent to all clients. COM Clients will not receive events for hooks enabled by Delphi 5.0 Customisation clients.

The following object is passed into each OnHook event as a parameter, it contains a copy of all the relevant data in Enterprise at the point the hook was reached:-

```
ICOMEEventData
  CostCentre : ICOMCCDept
  Customer : ICOMCustomer
  Department : ICOMCCDept
  GLCode : ICOMGLCode
  Job : ICOMJob
  MiscData : ICOMMiscData
  Stock : ICOMStock
  Supplier : ICOMCustomer
  Transaction : ICOMTransaction
    thLines : ICOMTransactionLines
      thCurrentLine : ICOMTransactionLine
      thLine : Array of ICOMTransactionLine
```

Depending on the specific hook various parts will be enabled and disabled, and specific fields will be available for writing to. Details of what each hook has available can be found in the on-line help file.

The OnClose event executes when Enterprise is closed, its purpose is to allow the COM Client to release its reference to the COM Object. Allowing Enterprise to close normally. If the client does not correctly release its reference then Enterprise might not be able to shutdown correctly.

EntFuncs.Dll

What Is It?

EntFuncs.Dll is a wrapper DLL for the Enterprise OLE Server which publishes several of the core functions used by the Exchequer Enterprise Add-In for MS Excel. It originally shipped with the Enterprise ODBC Kit, but is also now available for users of the Toolkit DLL.

What Can It Do?

The following functions are available:-

EntDefaultLogin

Allows a default User Id and Password to be set programmatically to avoid the login dialog being shown by the OLE Server.

EntCustValue

Returns Customer/Supplier History values.

EntGLValue

Returns General Ledger History values.

EntStockQty

Returns the In Stock quantity for Stock and Locations.

EntStockValue

Returns Stock History values.

EntFuncsShutDown

This function can be used to programmatically shut down the OLE Server.