IMEX Error Collection

Main...

IMEX ERROR COLLECTION

The 'IMEX Error Collection' is a class which makes the unattended functionality possible, it even removes the need for any error trapping code in the client application.

Overview
Registration

Methods
Add Method
Errors Method
Refresh Method

Properties
Action
Count
Source
Number

Action

Description The Action is useful to retrieve the current and most up to date Error Action within the Collection

Usage String$ = ERRS.Action

Data Type String *(default)*

Comments Action acts as a name placeholder for the running function,   sub-procedure or program.

Count

Description	The Count Property is used to retrieve the current error count that is held within the Collection
Usage	*iVal%* = ERRS.Count
Data Type	Integer
Comments	This property is used to return the current error count.

Overview

When would you use the IMEX Error Collection?

The IMEX Error Collection lends itself to various implementations where Ole Servers require automated unattended actions.   One possible application of this class is to handle and store errors without halting the current process and reduce the need for client application/programmer to add   "in-line reactive error handling".

This isolates the error routines to a specific area within your application and available at any time, even after the process has completed.

In a programming situation there might be 10,000 transactions to process but it does not matter if a few, or indeed all fail because the action must be completed.   In this scenario the IMEX Error Collection stores the transients to either memory or a log file for inspection by program or visual status results for a user.

What are the benefits ol the IMEX Error Collection?

The 'IMEX Error Collection' as a class makes the unattended functionality of the 'IMEX Ole Import/Export' possible, it even removes the need for any error trapping code in the client application.

This Class Module is a simple but effective means of trapping run time errors or transients within a collection.   Interrogation can be carried out during, or after any action, enabling the unattended automation that some Ole Servers require.   This Class removes the overhead of lengthy error trapping routines, saving you hours of trial and error coding.   Something we could all do with!

This Error Collection is continually being enhanced and we hope to release a more advanced version in early '96.

How to Purchase

The source code for this Class Module is supplied free with Registration of any IMEX product and offers the benefit of adding deferred error handling to any of your VB programmes.   If you want the IMEX Error Class on it's own it can be Registered through SWREG for $10 (US), £ 7.50(UK).

Number

Description	The Number Property is used to retrieve the most recent error number that is held within the Collection. Each error numbers holds it's own related Description.
Usage	*lVal*& = ERRS.Number
Data Type	Long
Comments	This property is used to return the current error number.

Source

Description The Source Property is used to sub-categorise the Name Property.

Usage *String$* = ERRS.Source

Data Type String

Comments Use to return the current errors source, pinpointing the original location where the error occurred. This an optional Property that can be ignored, however when used with the Name Property it is possible to give an more fine tuned error locating method.

Add Method

Description The Add Method 'adds' an error to the collection as it's name implies.

Usage vErrVal = ERRS.Add(Number&, Description$, [Source$], [Action$])

Data Type Variant *(Array)*

Comments The array that is returned *(held in a variant)* is made up of the following parts:

```
vErrVal(x%)
x% = 0  Number as a Long
x% = 1  Description as a String
x% = 2  Source as a String
x% = 3  Action as a String
```

Errors Method

Description	The Errors Method returns a variant which itself contains a two dimensional variant array.
Usage	*vArray* = ERRS.Errors
Data Type	Variant Array

Comments

After the array is returned the 'IMEX Error Collection' can be cleared by the Refresh Method.

Array Contents vArray(x%,y%) - values within the array

Dimension x% = the unique error event number related to the 'Count' at that snapshot in time.

Dimension y% = consists of four variants:-

0	*Error number (long)*
1	*Error description (string)*
2	*Error source (string)*
3	*Error action (string)*

Number	*A long value (contained in a variant) which indicates an error number.*
Description	*This is quite simply the explanation of the error in English.*
Source	*The source of the error*
Action	*The action is what was being processed at the time of the error.*

Refresh Method

Description The Refresh Method simply clears the contents of the collection an resets the Count Property back to 0.

Usage ERRS.Refresh

Comments Use this method to clear all errors held within the collection/class.

Registration

Registration via CompuServe

To register the IMEX Ole Server use CompuServe, go to the shareware registration forum (GO SWREG) and select the REGISTER SHAREWARE option. Our registration ID is 8760
Your CompuServe account will be debited automatically. Full instructions are given in the SWREG forum.

Registration via Snail Mail

Not available for this product

CompuServe Registration Prices

IMEX Error Collection + source code	$10 (US)

License and source code is available **Free** with registration of the main 'IMEX Ole Server' product.

Registration entitles you to use freely distribute the IMEX Error Collection with your own applications. However if you wish to resell any 'IMEX' product as part of a development tool, require site licenses or source code please contact Solugistics Ltd. and we will be delighted to arrange a special agreement.

Description

Description	The Description Property describes the current error in plain English using your own terms and understanding for the application's user.
Usage	*String$* = ERRS.Description
Data Type	String
Comments	This could be the actual returned language *(development environment)* error description or redefined description to aid any user with understanding what the error is and suggest possible actions to be taken.

Glossary

A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z

N

name placeholder

name placeholder
This is a unique descriptor that   names the function,   proceedure   or indeed the position whithin the running program.