

SYSDECO TOOLS

QBE Vision Developer



QBE Vision is a comprehensive development tool for all Windows platforms. QBE Vision supports the development of a full range of systems,

from the simple to the very advanced.

QBE Vision consists of four integrated modules which are appropriate to different categories of users.

Since the end of the 1980s QBE Vision has undergone a continual process of development, in close co-operation with software houses producing large application systems. This has resulted in a powerful tool with the capability of producing applications with the required functionality both quickly and efficiently.

From prototyping to completion

QBE Vision development is based on starting with a user interface as a prototype for the final application. This user interface is then extended by linking code to the various events which can be triggered by the different user interface components. The resulting programs are ready for immediate execution.

Standard programming language

The programming language used in QBE is Basic, a language which is becoming a standard for Windows programming. QBE Vision contains about 500 built-in functions for handling database access and the interface with the user.

The Basic language used in QBE has also been extended to encompass object-oriented mechanisms such as encapsulation, classes, inheritance and virtual functions.

This functionality is implemented such that users can take it into use once they feel ready to advance to using object-oriented technology.

A built-in compiler

Completed program sections can be run through an interpreter, which is especially useful during the development/test phase of a project. Fully completed applications can be compiled to so-called P-CODE for quicker execution. Such ready-compiled applications can be run without alteration on all Windows platforms, both 16 and 32 bits. No code changes or re-compilation are necessary.

Fast database access

In contrast to many tools available on the market today, QBE directly calls the various database suppliers' own programming interfaces. This results in a reduced overhead, which again gives rise to faster database access. In the case of database suppliers who do not offer their own programming interface, ODBC is used.

Using QBE's own database access from other tools

QBE's own database interface has been made available as an ActiveX Automation server. This means that it can be used from other tools, such as Microsoft Office.

Dynamic code generation

Advanced programmers can take advantage of QBE's functionality for generating Basic code during the execution of an application. The generated code can then be compiled and executed while the application is still running.



SYSDECO

Re-using code

Application code to be made available for re-use in other applications can be stored in object libraries. This results in smaller applications and large savings on maintenance. Any changes to the code in such object libraries will be immediately effective in all applications using it, without the need for these applications to be changed or even re-compiled.

Re-using the user interface

As well as application code, selected components from the user interface, together with their associated underlying event-handling code, can be stored in object libraries. This gives applications the possibility of sharing common user interface components and behavior. This is important with regard to standardization, as well as giving large savings in maintenance costs.

This technology makes it possible to assemble an application very quickly by putting together ready-tested components from object libraries.

During the execution of an application it is also possible to dynamically load different components, depending on certain events. Doing so will often reduce the application's execution time, because only those components which are needed at any one point of time will occupy memory. Once components are no longer required, they can be released and their memory freed.

Any changes to a user interface component in an object library will immediately be effective in all applications using the component, without the need for these applications to be changed or even re-compiled.

The repository

QBE Vision has its own repository which can be used when developing large projects. Amongst other things, the repository provides "native language support" allowing all language dependent elements, such as leading texts, display formats, sizes, legal values, messages, etc. to be stored in the repository, instead of in the application itself.

Translation to other languages can be performed simply by pressing a button in QBE to select the new language. All changes to elements in the re-

pository are reflected in the applications using them. Thus, all language dependent elements in the object libraries and applications are automatically translated to the chosen language.

The repository can also supply information as to "what is being used where". This is important in large projects with many developers.

Using ActiveX technology

ActiveX technology offers many possibilities for integrating applications and components from third party suppliers. Irrespective of how, and in which programming languages, components have been implemented, they can be used in QBE applications. It is, for example, possible to call the functions made available in many of today's spreadsheet programs directly from QBE applications.

Ready-made components with a user interface and callable functions can be used in QBE as if they are an integrated part of the product.

Developing ActiveX components

As well as being able to use ready-made ActiveX components and functions, with QBE it is also possible to develop one's own functions and components which can be used in applications developed both with QBE and other tools.

What about existing applications?

With QBE Vision 6.0 it is very easy to convert existing applications, which advantageously could have been ActiveX Automation servers or ActiveX components, without any re-writing.



SYSDECO