

APPENDIX A

DESCRIPTION OF *AM for Windows* SOFTWARE ELEMENTS

1. Modules

The module is the largest editable unit in a collection of procedures in a single application file. In *AM Builder*, application consists of a single module, while in *AM Enterprise*, application can consist of a base module and a number of other *dynamically linked modules* (DLMs). DLMs are normally used in the multi-tasking applications, which can be run more than one module simultaneously and are capable to support a communication between tasks.

2. Procedures

A procedure contains one or more single line statements that can call other procedures or perform AM commands. Procedures help to define the basic structure of a module.

3. Commands

A command is a request to perform some action. For example, there is the **:Assign** command, which assigns a value to a variable, and the **:Test** command, which tests the value of a variable or expression.

Commands can be inserted by selecting the appropriate command from a Commands menu. For some commands, AM displays a dialog box to help the developer complete the command.

4. Variables

AM supports text and numeric variables. The variables can be simple single value variables, multi-dimensional arrays, or multi-column tables.

Examples are:

Text variable	→	LeanName\$
Numeric variable	→	LeanNumber
One dimensional text variable	→	LeanTable{LeanName\$}
One dimensional numeric variable	→	LeanTable{LeanNumber}
Multi-column variable	→	LeanTable{LeanName\$, LeanNumber}

5. Windows

A *Window* is how a module can communicate with the user. *Windows* can display information to the user and receive information from the user. AM supports standard *Windows* controls such as entry fields, list boxes, push buttons, radio buttons, and so on. Additional graphical controls (such as picture buttons and bar charts) are supplied with AM in the UDC and GCL AMX libraries. Developer's own controls can also be written by using the AMX Development Kit.

6. Functions

A function is a ready-written routine that can be called from a module to perform some common action such as date arithmetic, financial and scientific calculations, text manipulation, and so on. Here are two example function calls:

Beep (400, 2) → (*sounds a beep at 400Hz for 2 seconds*)

Left\$("YESNODON'TKNOW", 3) → (*returns the 3 left most characters, Yes*)

A large number of functions are built into AM. Additional functions can be made available by loading one or more of the AMX libraries supplied with AM. Own libraries of functions can also be written by using the AMX Development Kit.

7. Menus and Buttons

A menu bar can be displayed by a module to provide an easy way for users to select actions. The menu bar can contain any number of menu headers. Items in menus can have short-cut keys assigned. For example, **Ctrl + O** is the key for the File/Open menu item.

In addition, a Toolbar containing buttons that activate menu items can also be created. Buttons can be assigned to commonly used menu items with their action and appearance controlled using AM Toolbar functions. Short descriptive messages, known as Tool Tips, can also be attached to buttons. At runtime, when a mouse pointer hovers over a button the Tool Tip is displayed.