

**PROGRAMMING GUIDELINES:
NAMING CONVENTIONS**

by Jim Dolson
and Greg Lang

Purpose

The purpose for proposing these programming guidelines is four fold. In order of importance, they are:

1. To make it easier to work on a multi-programmer project
2. To aid in the debugging process
3. To aid in the maintenance process
4. To aid in any future porting process

Through the thoughtful use of naming conventions, anyone can look at a piece of code and instantly tell what controls are involved and the data type and scope of any variable. The benefit of this in multi-programmer projects, debugging, maintenance, and porting can not be overstated.

Programming Language Considerations

Adding to the confusion is the fact that each language vendor has their own recommendations for naming conventions. Most programming languages support the same basic group of data types and objects. We wish to create consistency across programming languages for these common elements, so that a check box in an application will have the same name regardless of the programming language used..

When a language supports a data type or object that is not supported in other languages, the following general guidelines can be applied to derive a new convention for that data type of control.

Naming Conventions: Variables & Constants

All variable and constant names are of the form: **{type}{scope}{root}**

Root Names

The root portion of a variable name is a short but clear description of what the variable contains. The first letter is always capitalized and trailing letters are never capitalized (unless the variable name root has a commonly recognized abbreviation, such as SSN). If the variable name root contains two or more words, then the first letter of each word is capitalized, the remainder are lower case, and there is no space between words. No underscores should appear anywhere within the variable name root.

The root name of a constant should also be a short but clear description of what the variable contains. All letters in the name are capitalized. If the constant name root contains two or more words, then use the underscore '_' character to separate the words.

For example:

<u>Use ...</u>	<u>Instead Of ...</u>
Age	a or age
FirstName	fn or fname
WorldPopulation	WPOP
SSN or SocialSecurityNumber	ssnum or ss_num
PI	pi or Pi

Typing

The type of a variable can be determined by examining the first letter of the variable name. The following table lists types and their corresponding single letter identifier.

Boolean	b
Integer	i
Long Integers	l
Single Float	f
Double Float	d
Currency	c
String	s
Variant	v

For Example:

<u>Use ...</u>	<u>Instead Of ...</u>
iAge	Age
sFirstName	FirstName
fWorldPopulation	WorldPopulation
fPI	PI

Scope

The scope of a variable can be determined by examining the second letter of the variable name. The following table indicates scopes and their corresponding single letter identifier.

Global	g
Module	m
Local	{l}

Note that for local scope, you may either use an 'l' or not use a scope identifier at all. If no scope identifier is present, the variable is assumed to be of local scope.

For Example:

	<u>Variable w/o Scope</u>	<u>Variable w/Scope</u>
	iAge	iAge or ilAge if local
	sFirstName	smFirstName if module
	fWorldPopulation	fgWorldPopulation if global
scope	fPI	fgPI

Declaration

All variables are to be explicitly declared. This will force you to explicitly declare each variable. If you attempt to use a variable in your program that you have not explicitly typed, the compiler will flag it for you.

Naming Conventions: Controls

All control names are of the form **{type}{root}**

Root Names

The root portion of a control name is a short but clear description of what the control contains. The first letter is always capitalized and trailing letters are never capitalized (unless the control name root has a commonly recognized abbreviation, such as SSN). If the control name root contains two or more words, then the first letter of each word is capitalized, the remainder are lower case, and there is no space between words. No underscores should appear anywhere within the control name root.

Typing

The type portion of a control name is a three letter abbreviation indicating what kind of control it is. The following table lists controls and their corresponding three letter identifier.

Check box	chk	Combo box	cbo
Command button	cmd	Data	dta
Directory list box	dir	Drive list box	drv
File list box	fil	Form	frm
Frame	fra	Grid	grd
Horizontal scroll bar	hsb	Image	img
Label	lbl	Line	lin
List box	lst	MDI form	mdi
Menu	mnu	OLE	ole
Option button	opt	Picture box	pic
Shape	shp	Text box	txt
Vertical scroll bar	vsb	Timer	tmr

For example:

Check boxes on a 1040 to indicate whether the filer is over 65, blind, and/or disabled:

chkOver65, chkBlind, chkDisabled

A label that prompts the user to enter their first name:

lblAgePrompt

The OK button on a form:

cmdOK

A list box containing the employees in a department:

lstDeptEmployees

Two option buttons to signify male or female (but not both):

optMale, optFemale