

Personal Designer
User Programming Language
(UPL)

Revision 6.0

User Reference Guide

Chapter 3

Functional Listing

Functional Listing

Program Structure Statements	3-4
Declaration Statements	3-4
How Control Statements	3-4
Compiler Directives	3-5
Assignment Statement	3-5
Database Access	3-5
Database Access Statements	3-6
Database Access Intrinsics	3-6
Input/Output	3-7
File Input/Output	3-7
Window Input/Output	3-8
Window Input/Output Statements	3-9
Window Input/Output Intrinsics	3-10
Graphics Intrinsics	3-11
User Interface Intrinsics	3-11
Geometric Intrinsics	3-12
Arithmetic Intrinsics	3-14
Trigonometric Intrinsics	3-15
String Handling Intrinsics	3-15
Data Conversion Intrinsics	3-15
Operating System Intrinsics	3-16

Functional Listing

This chapter groups UPL statements and intrinsics by the tasks which they perform and gives a short definition of each one. (Intrinsics are standard functions and procedures that are already defined in the code and are available for use as part of UPL.) You can read through this chapter while you are learning UPL to get a good overview of its facilities. Later, while you are programming, you can use the chapter to find the name of the statement or intrinsic which performs a particular function. Then use the alphabetically-arranged reference in Chapter 4 to get specific information on syntax, operands, arguments and other details.

The categories used in this chapter are shown below, in the order they appear.

Statements

- Program Structure
- Declaration
- Flow Control
- Compiler Directives
- Assignment
- Database Access
- Input/Output

Intrinsics

- Database Access
- Input/Output
- Graphics
- User Interface
- Geometric
- Arithmetic
- Trigonometric
- String Handling
- Data Conversion
- Operating System

Functional Listing

Program Structure Statements

These statements define the structure of the program.

Proc/ End Proc	defines the boundaries of a procedure in UPL. There must be at least one procedure in a UPL program, the Proc Main.
Func / Return / End Func	defines the boundaries of a function and the type of value it returns.
Group/End Group	defines the variable's storage or access types to be global, which means they are accessible to all procedures and functions. If used in a program, this section must appear first.

Declaration Statements

These statements declare the data and aggregate variable types. All variables must be declared before they are used in a program.

Boolean	declares Boolean variables.
Coord	declares coordinate variables.
Integer	declares integer variables.
Real	declares real variables.
String	declares character string variables.
File	declares file variables.
Const	declares a named constant.. Is used together with the above keywords.

Flow Control Statements

These statements control the path of execution for the program. UPL supports three basic flow control constructs: sequence, branching, and iteration.

If/Then/Else/Elseif/Endif	supports conditional branching.
Loop/EndLoop	supports iteration.
Exit	allows escape from a Loop or an If construct.
GoTo	transfers control to a specified point unconditionally.

Functional Listing

UPL programs can be partitioned into modular "building blocks." These can perform specific tasks and be used repeatedly. The blocks may be procedures, functions, or other UPL programs. Flow of control is passed to these blocks when they are executed and then returned to the program when they are finished.

Process executes another UPL program from within the current one.

Procedure/function call invokes procedure or function.

Return returns control from a procedure or function call.

Sleep suspends execution for a specified number of instructions.

Compiler Directives

Directives control aspects of the compilation process. They must be preceded immediately by a '\$'.

Codesize controls how much external memory is allocated for UPL code.

Include inserts external code files onto your program.

Assignment Statement

... = ... statement assigns values to variables.

Database Access

A drawing or part is stored in the part database. These are special purpose files which store all the data defining the entities. A UPL program may query the database and modify or add entities to it.

Database Access Statements

These statements allow the program to manipulate database entities. Line, arc, point, text, and string entities are fully supported. Note that string entities and character strings are not the same. String entities hold connected line segments. They are a type of entity used in Personal Designer. Character string types are a type of variable that holds a sequence of one or more characters.

Functional Listing

Verify	returns information about an entity.
Insert	inserts an entity into the database.
Modify	modifies an entity in the database.
Erase	erases an entity from the database.

Database Access Intrinsic

The following intrinsic allow you to query the database for non-graphical information attached to the entities.

FindProp	returns the value of a specified property and the entity on which it occurs.
SetTagField	sets entity tag information.
GetTagField	returns entity tag information.
MibTag	returns the entity tag when given an MIB number.
TagMib	returns an MIB number given an entity tag.

The following intrinsic procedures allow you to move, rotate, and scale entities just as one would with the appropriate Personal Designer command.

MirEnt	mirrors entities according to a transformation matrix.
MirEntCopy	mirrors and copies entities according to a transformation matrix.
MovEnt	moves entities according to a transformation matrix.
MovEntCopy	moves and copies entities according to a transformation matrix.
RotEnt	rotates entities according to a transformation matrix.
RotEntCopy	rotates entities according to a transformation matrix.
SciEnt	scales entities according to a transformation.
SciEntCopy	scales and copies entities according a transformation.
StrWide	inserts a string of entities about a given string.

Functional Listing

Most UPL programs involving database access may be written using the above statements and intrinsics. See Appendix G for more information on accessing the database.

Input / Output

Input/Output takes two forms in UPL: File and Window. Files can be used to hold data a UPL program will use. Windows are used to interact with the user. File and Window I/O uses statements and some special intrinsic procedures.

File Input / Output

Files can be text or binary. They can be accessed sequentially or randomly. These are the statements which manipulate files.

Open	statement opens or creates a file.
Read	statement reads data from a file.
Write	statement writes data to a file.
Close	statement closes a file and saves its contents.
Delete	statement deletes a file.

There are a number of intrinsic procedures that perform input/output to binary files and allow the program to work with amounts of data larger than 32,767 bytes. These are:

ReadCArray, ReadIArray, ReadRArray	for reading data.
WriteCArray, WriteIArray, WriteRArray	for writing data.

Window Input / Output

Window I/O is generally the interaction between the UPL program and the user. Just as in Personal Designer, a UPL program interacts with the user by prompting for and returning information. This information is output or echoed in windows. (Windows are rectangular regions of the display screen.)

Functional Listing

Personal Designer has one graphics window and 20 alphanumeric windows. These are all accessible to a UPL program. The graphics window is used to display the geometry and interact with the user as he digitizes it. Alphanumeric windows are used to interact with the user as he types on the keyboard or picks icons from a menu. Of the 20 alphanumeric windows, 10 are available for use by the UPL program. The remaining windows are reserved by Personal Designer to output data, however, their appearance, location, size, and other characteristics can be controlled by a UPL program.

Below is a list of the alphanumeric windows and their default priorities.

Window Number	Default Priority	Description
1	11	UPL window 1 and Personal Designer command/prompt window
2	7	UPL window 2
3	8	UPL window 3
4	9	UPL window 4
5	10	UPL window 5
6	11	UPL window 6
7	12	UPL window 7
8	13	UPL window 8
9	14	UPL window 9
10	15	UPL window 10
11	6	message window
12	17	warning message window
13	18	error message window
14	6	X, Y coordinates of cursor location window
15	20	status window
16	6	help window
17	6	fast data display window
18	6	DOS window
19	24	reserved
20	25	reserved

Functional Listing

The graphics window has the lowest priority of all windows, and the on-screen icon menu has the highest. This means that any window can cover the graphics window but none can cover the on-screen icons.

Window Input / Output Statements

These statements affect UPL windows 1-10 and the graphics window.

Window	defines the size and placement of a window.
Clear	clears a window.
Accept	statement accepts data from the user and echoes it to a window.
Display	statement displays text in an alphanumeric window.
Print	statement prints the value of data in an alphanumeric window.
Send	statement sends commands to Personal Designer and echoes them in an alphanumeric window.

Echo controls the echoing of data in the windows.

The **Accept, Display, Print, and Send** commands perform I/O using the windows specified in their respective system variables: **AccptWin, DisplayWin, PrintWin, and SendWin**. These all default to window one at the beginning of a program.

When using **Window and Clear** statements, specifying window 11 refers to the graphics window. To clear alphanumeric window 11 use **CLEAR 11** or make a call to **AWinClear(11)**. This is to maintain compatibility with UPL version 1.

Each UPL window has a cursor which starts in its upper left-hand corner. Subsequent window statements will leave the cursor after the last character output. UPL remembers the last cursor position for each window. The intrinsic routine **PutCur** allows the cursor to be moved to a specified position. Windows will automatically scroll when the cursor reaches the bottom of the window.

Functional Listing

Most UPL programs can be written using the above statements on UPL windows 1-10 and the graphics window. Advanced programmers who want more control over windows and access to windows 11-20 should use the intrinsic procedures listed below. These allow control of all windows and allow the program to perform tasks not possible with the Window and Clear statements.

Window Input / Output Intrinsics

DefineAW	defines or returns information about alphanumeric windows.
AWinClear	clears the specified alphanumeric window.
TextColor	sets the color of text to draw in alphanumeric windows.
RowColAW	returns the number of rows and columns in an alphanumeric window.
RowColToPix	maps row and column coordinates to pixel coordinates.
PixToRowCol	maps pixel coordinates to row and column coordinates.
GWinClear	clears the specified graphics window.
GetCur	returns current cursor position.
PutCur	sets cursor position.

When any windows overlap, one will cover the other. Their new appearance will then depend upon their window numbers and priorities. When two UPL windows (number 1-10) overlap, the "current window" will cover the other. The current window is the one most recently used by the **Clear, Accept, Display, Print, or Send** statements. When any other combination of windows overlap, the window with the highest priority will cover the other. The window with the lower priority will then shrink to the largest remaining rectangular region within its original bounds. The contents of a window which is covered by another window is lost.

Functional Listing

The graphics device page is the area of the display in which all interaction takes place. This includes the Personal Designer graphics window, the Personal Designer onscreen menus, the Personal Designer command windows and any other Personal Designer windows visible at the time. If you are running under DOS, this is the whole display screen. If you are running under UNIX, it may be either the whole display screen or a window maintained by the operating system.

The origin (0,0 point) of the Pixel coordinates is at the lower left corner of the graphics device page. The origin of the row and column coordinates is at the upper left corner of the graphics device page.

Graphics Intrinsics

These intrinsic functions allow the UPL program to control certain aspects of the graphics and text displayed in the windows.

HiliteEnt	highlights entities on the screen.
RpntEnt	repaints specific entities in the data base.
Draw	draws graphics on the screen without adding them to the database.
DrawText	draws graphic text on screen without adding it to the database.
GText	draws alphanumeric anywhere on the graphics screen.
ShadeColor	returns a color index number when given a color and shade value.
FillPoly	fills a polygon with solid shade of color.
SetLayer	sets status of layers to on or off.
GetLayer	returns status of layers.

User Interface Intrinsics

These intrinsic procedures aid in customizing Personal Designer by making your UPI, programs act like Personal Designer commands. This includes routines to work with on-screen icons and command modifiers. For more information, see "Writing Personal Designer Commands " in Appendix H.

Functional Listing

DefineModifier	defines modifier information.
AskModifiers	gets modifier values from the user.
GetModifier	gets specific modifier information.
SetMenuInfo	sets up on-screen menu information.
GetMenuInfo	gets on-screen menu information.
HilighMenu	highlights an on-screen menu.
FindMenu	returns the menu item containing a given point.
MenuCmd	sends menu commands to the menu command processor.
SetHelp	assigns help index for the on-line help system.
GetHelp	allows entry to the on-line menu system.
GetEnt	returns MIB pointers of digitized entities.
GetEnd	returns MIB pointers of entities whose ends are digitized.
GetDig	returns coordinates of a digitized position.
GetC	returns a character or pixel coordinate position.
InputStr	inputs characters to input buffer.
FlushInput	flushes the input buffer.
LastDig	returns the last digitized position.
BigMibList	returns an unlimited number of the MIB pointers of the digitized entities from a GetEnt call.
EntMask	determines what kind of entities will be accepted by the GetEnt call.

Functional Listing

Geometric Intrinsic

Geometric intrinsic procedures and functions allow the user to find geometric relationships between entities in the database.

Ang3P	finds the angle between two intersecting lines in space.
EntIntOf	finds the intersection of two entities closest to a given point.
LinIntOf	finds the intersection point of two lines.
EntPntOn	determines which point on an entity is closest to a given point.
PntPrp	returns a point perpendicular to a line in model space.
PntPrpV	returns a point perpendicular to a line in view space.
PolyArea	finds the area of a polygon.
PolyWin	determines if a point is on the inside of a polygon.
MapVM	maps a coordinate from view to model space.
MapMV	maps a coordinate from model to view space.
MapCPLM	maps coordinates from a CPL to model space.
MapMCPL	maps coordinates from model space to a CPL.
MapFrom	maps transformed coordinates to model coordinates.
MapTo	maps model coordinates to transformed coordinates.
MapTT	maps coordinates directly from one transform to another.
Map2Px	maps a point from view to pixel coordinates.
Map2PxN	maps arrays of points from to pixel coordinates.
MaPix2	maps pixel coordinates to view space coordinates.
MaPix2N	maps arrays of pixel coordinates to view space coordinates.

Functional Listing

Nullrtransform	creates a null transformation matrix. This is useful for matrix initialization.
Mat3p	creates a transformation matrix given three points.
Transpose	procedure transposes a transformation matrix along its diagonal elements.
GetCPL	gets the transformation matrix for a given CPL.
GetView	gets the transformation matrix for a given VIEW.
RotMat	returns a transformation matrix describing a rotation.
RotPnt	rotates a set of points given a transformation matrix.
MirPnt	mirrors points about a plane.
Vross	returns the cross (vector) product of two vectors.
VDot	returns the dot (scalar) product of two vectors.
VLen	returns the length of a vector.
VUnit	returns a unit vector parallel to a given vector.

Arithmetic Intrinsics

These intrinsics supply some basic mathematical capabilities.

Abs	returns the absolute value of a number.
Ln	returns the natural logarithm of a number.
Log	returns the base 10 logarithm of a number.
Max	returns the maximum value of a given set of numbers.
Min	returns the minimum value of a given set of numbers.
Rnd	returns a random number.
SqRt	returns the square root of a number.
ModI	returns the remainder of an integer division.
ModR	returns the remainder of real division.
GetBit	sets a bit in a binary bit table.
SetBit	clears a bit in a binary bit table.

Functional Listing

Trigonometric Intrinsic

These intrinsics provide the basic trigonometric functions.

Cos	returns the cosine of an angle.
Sin	returns the sine of an angle.
Tan	returns the tangent of an angle.
ACos	returns the arccosine of a number.
ASin	returns the arcsine of a number.
ATan	returns the arctangent of a number.
ATan2	returns the arctangent of a ratio.
Pi	returns the value 3.141593
TwoPi	returns the value 6.283185

String Handling Intrinsic

Extract	function extracts a substring from a given string.
Index	finds the index of a substring in a given string.
RmvChr	removes from a string all occurrences of a given set of characters.
UpperCase	converts lower case letters to uppercase in a string.

Data Conversion Intrinsic

These intrinsics convert data from one data type or format to another. These can be very helpful since UPL requires all parts of an expression to have the same data type.

Boolean	converts a value to a Boolean value.
Integer	converts a value to an integer value.
Integer4	converts a value to an integer4 value.

Functional Listing

Real	converts a value to a real value.
String	converts a value to a string value.
Coord	converts three real expressions into one coordinate value.
DigStr	converts a coordinate expression into the correct format for passing to the Send statement.
Char	returns an equivalent character for an ASCII value.
ASCII	returns the ASCII value of the first character in a string.
CntrlToNum	converts non-printing characters to Personal Designer format.
NumToCntrl	converts non-printing characters from Personal Designer format.
DegRad	converts a value from degrees to radians.
RadDeg	converts a value from radians to degrees.

*Operating System Intrinsic*s

These intrinsic procedures and functions allows access to Personal Designer and the operating system (OS) to perform certain key functions.

DOS	allows execution of a DOS level command. It also allows temporary escape to DOS until the command EXIT returns control to the UPL program.
EnvVar	obtains the value of an OS environment variable.
SysVarI	returns the value of Personal Designer system variables with integer data type.
SysVarI4	returns the value of Personal Designer system variables with integer4 data type.
SysVarR	returns the value of Personal Designer system variables with real data type.

Functional Listing

SysVarS	returns the value of Personal Designer system variables with string data type.
Date	returns the system date in the format: MM/DDIYY
Time	returns the system time in the format: HH:MM:SS.SS.
Size	returns the size of a file in 128 byte blocks.
Size4	returns the size of a file in bytes.
Exist	checks to see if a file exists.
AddFnExt	allows the program to add an extension to a file name.
Diffn	gets directory information about files on the system.
FindFn	determines in which directories a file with a given name exists.
IDiskFree	returns the amount of free disk space on the system as an integer4 value.
DiskFree	returns the amount of free disk space on the system as a real value.
MemAvail	returns the amount of data on the UPL data stack.
PageInfo	returns information about a particular page in a graphics device.
Product	returns information about the system running the UPL program. This includes: product version number, graphics device id number, etc.
AccessCode	returns a unique access code for a given key and serial number.
GetSerialNum	returns the serial number of the system's software protection device.
GetKbdChar	gets the next character from the keyboard.
MouseInput	gets the location, status, and resolution of the input device.
Device	allows direct interface to device drivers.
AsmInt	allows software interrupt. (DOS only)