FPDL Ref

FRM file language **Language Reference**

Header:

FORM-FAMILY *FamilyName*
FORM-NAME *FormName*

Main macro format:

```
Macro Main
{
        Operations;
}
```

Macro Format:

```
Macro macroName
{
        atomic operators;
}
```

Data definition statements

```
Define Struct StructTypeName      //  Struct definition:
{
        Structs, Strings, Numbers, Lists
}
```

Data decraration statements

```
String StringName;
Int    IntName;
StructTypeName      StructName;       //  Struct Instantiation:
List   ListName;    // List instantiation:
```

Special constants

%Date% : Date in mm/dd/yyyy format (All dates are relative to the local system time)
%Day% : Day in Mon, Tue, Wed format
%Time% : Time in 24hr, 16:24:02 format
%FormFamily% : The name of the form family being processed
%FormName% : The name of the form type being processed
%JobIdentifier% : The Job identifier of the current job

Data navigation statements

Prev(*param*) :Atomic operators; *param* may be a number (repeat factor)
Next(*param*) :or a String constant or variable (repeat until cursor value is

Left(*param*) :equal to *param*)
Right(*param*)

Advanced Param – *BLOCKEND, BLOCKSTART, MAX, MAX*

Home; : Moves the cursor back to the home position (by default the beginning : of the file.
SetHome; :Sets the home position to the current cursor position.
PushHome; : Saves the current Home position on a stack
PopHome; : Sets the Home position to the top value on the stack (if the stack is : empty, does nothing)

Flow control statements

Repeat(*n*) {*operations*} : repeats the operations within the curly brackets *n* times

If ("*textstring*") executes the statements if the value at the cursor
{ *operations* } position is equal to *textstring*, or those in the next brackets if not.
Else
{ *operations* }

Until("*textstring*") : executes the statements in the brackets while the value of the cell at the
{ *operations* } : current cursor position is not equal to *textstring*

Permutations of the If and Until loops should also be implemented, where instead of a literal string constant a variable can be substituted, or as a third case, where an equality expression is used in place of the string constant, and the cursor value is not compared.

Done; : exits the macro

ID ; call other macro

Data input/output

Get(*varName*); : Sets the value of *varName* to the value at the current cursor position

Print("*constant string*");
Print(*varName*);
Print(*Print list*)

Arithmetical operations

= : set left hand variable to the value of the right hand side (either a variable or constant (String or Number)

+ - / *

ListName = varName; // List assignment:

Logilcal operations

==, **!=** : Equal and not Equal logical operators.

String Type Operators:

IsText	(alphabetic characters and standalone numerals)
IsNumeric	(only numeric characters)
IsAlpha	(only alphabetic characters and ".")
IsDate	04/22/2xxxx, 04/22/xx
IsDollar	124.45
IsSSN	123-45-8594
IsBlank	(contains no characters)

These operators will return true if the string matches the listed format and will be called with the syntax:

*StringName.*IsPercent