

EDL Help
EDL Grammar

=====

EDLProgram ::= *Statement**

Statement ::= *CommandName* ["(" *Parameters* ")"] ";"
| "Element" "(" *Name* "," *TypeName* ")" *Body*
| "List" "(" *Name* "," *TypeName* ["," *ListLen*] ")" *Body*
| "Switch" "(" *Name* ")" *SwitchBody*

Body ::= *Statement*
| "" *Statement** ""
| "SELF" ";"

SwitchBody ::= "" *SwitchEntry** ""

SwitchEntry ::= *SwitchLabel*+ "" *Statement** ""

SwitchLabel ::= "case" *Expression* ":" ;must be a constant expression
| "default" ":"

Parameters ::= *Parameter*
| *Parameter* "," *Parameters*

Parameter ::= *Expression* | *string*

Expression ::= *Expression* "|" *AndExpr*
| *AndExpr*

AndExpr ::= *AndExpr* "&" *AddExpr*
| *AddExpr*

AddExpr ::= *AddExpr* ("+" | "-") *MulExpr*
| *MulExpr*

MulExpr ::= *MulExpr* ("*" | "/" | "%") *UnaryExpr*
| *UnaryExpr*

UnaryExpr ::= ["+" | "-" | "~"] *Term*

Term ::= *number* | *charConstant* | *identifier*
| "(" *Expression* ")"
| "Len" "(" *Name* ")"

ListLen ::= *Expression*
| "ZeroBasedCount" "(" *bitCount* ")"
| "OneBasedCount" "(" *bitCount* ")"
| "ZeroTerminator" "(" *bitCount* ")"
| "Unbounded"

CommandName ::= *string* ;see below for a list of commands

TypeName ::= *string* ;name of an element type

Name ::= *string* ;attribute name in the current element

Command summary

=====

Bool(*name*, *bitCount*);
SBits(*name*, *bitCount*);

UBits(name,bitCount);
Byte(name);
UByte(name);
Word(name);
UWord(name);
Long(name);
ULong(name);
ResourceRef(name,resType,bitCount);
VarResourceRef(name,typeAttr, bitCount);
PString(name);
CString(name);
PString(name,fieldLen);
CString(name,fieldLen);
String(name,len);
EvenPString(name);
OddPString(name);
EvenCString(name);
OddCString(name);
EvenPXString(name);
OddPXString(name);
Default(value);
Data(name,byteCount);
PadBits(bitCount);
PadByte;
PadWord;
PadLong;
PadBytes(fieldLen);
Align(byteCount);
Label(name);
LabelAttr(name);
OptionalEnd;
CalcBitsS(value,bitCount);
CalcBitsU(value,bitCount);