

```
-- file CmdScan.mesa
-- last modified by Sandman, August 30, 1977 10:46 AM
```

DIRECTORY

```
StreamDefs: FROM "streamdefs",
StringDefs: FROM "stringdefs",
CmdDefs: FROM "cmddefs",
IODefs: FROM "iodefs";
```

DEFINITIONS FROM StreamDefs, StringDefs, IODefs;

CmdScan: PROGRAM

```
IMPORTS StringDefs
EXPORTS CmdDefs =
BEGIN
```

ReadCmdString: PUBLIC PROCEDURE [input, name, switches: STRING] RETURNS [BOOLEAN] =

```
BEGIN
  i: CARDINAL ← 0;
  j: CARDINAL;
  activestring: STRING ← name;
  c: CHARACTER;

  name.length ← 0; switches.length ← 0;
```

```
--skip leading blanks --
WHILE i < input.length DO
  c←input[i];
  SELECT c FROM
    SP => NULL;
  ENDCASE => EXIT;
  i ← i+1;
ENDLOOP;
```

IF i = input.length THEN BEGIN input.length ← 0; RETURN[FALSE] END;

```
--parse command--
FOR i IN [1..input.length) DO
  c←input[i];
  SELECT c FROM
    '/' => activestring ← switches;
    SP,CR => EXIT;
  ENDCASE => AppendChar[activestring,c];
  REPEAT FINISHED => input.length ← 0;
ENDLOOP;
```

```
IF input.length # 0 THEN
  BEGIN
    j ← 0;
    WHILE (i+i+1) < input.length DO
      input[j] ← input[i];
      j ← j+1;
    ENDLOOP;
    input.length ← j;
  END;
```

```
RETURN[TRUE];
END;
```

ReadCmdStream: PUBLIC PROCEDURE [stream: StreamHandle, name, switches: STRING] RETURNS [BOOLEAN] =

```
BEGIN
  s: STRING ← [80];
  c: CHARACTER←SP;
  WHILE NOT stream.eofof[stream]
    DO c ← stream.get[stream];
    SELECT c FROM
      SP => IF s.length # 0 THEN EXIT;
      CR => EXIT;
    ENDCASE => AppendChar[s,c];
  ENDLOOP;
  RETURN[ReadCmdString[s,name,switches]];
END;
```

SkipExecCommands: PUBLIC PROCEDURE [stream: StreamHandle] =

```
BEGIN
```

```
name: STRING ← [40];  
switches: STRING ← [10];  
  
[] ← ReadCmdStream[stream,name,switches];  
END;
```

```
END.
```