Recognize invalid/unexpected characters with Perl

Today a colleague of mine faced a very weird problem.
While parsing XML output from an HP ILO into Perl, his code constantly broke with the message:

FILE.XML:123 parser error : PCDATA invalid Char value 1

While the message itself states clearly that there is an unexpected character value (Char value 1, ASCII SOH) on one hand, it doesn't tell the character position on the other.

Looking at the input string itself on the console, it wasn't obvious either:

<EVENT SEVERITY="Caution" LAST_UPDATE="08/03/2009 22:20" INITIAL_UPDATE="08/03/2009 22:20" COUNT="1" DESCRIPTION="POST Error: " />

So I proposed to add some lines to help identify the character position on the given input string, which was basically this:

```
@array = unpack("C*", $_my_input_var);
foreach (@array) {
printf("char "%s" is ord %sn", chr($_), $_);
}
```

This led to the following output:

char "" is ord 62

So looking at this we saw that ASCII char 1 (which is an unprintable character, it will be represented as A in some editors like vi) was the fifth character before the end of the string.

Well, basically the solution to this is to apply an additional input filter to remove ASCII char 1 like this:

```
$_my_input_var  =~ s/x01//g;
```

While this solves just _this_ problem, a more solid solution is to remove all non-printable characters as well, given the list of ASCII characters at http://www.asciitable.com/.

So a filter like this may apply, removing all non-printable characters, leaving just a few control characters 1x08 to 1x1F (Tab, Carriage Return, Line Feed and a few others) and the printable characters in it.

```
$_my_input_var =~ s/[x00-x08x0B-x1Fx7F-xFF]//g;
```