



```

949 949 # up other parse trees. Maybe should have tests for that, too.
950 950 DOT = ValueTerminal('.', 'dot')
951 951 ListSeparator = ValueTerminal(',', 'list-separator')
952 + ListSeparator.as_ew_allowed = False
952 953 RouteComponentMarker = ValueTerminal('@', 'route-component-marker')
953 954
954 955 #

```

@@ -2022,7 +2023,7 @@ def get\_address\_list(value):

```

2022 2023         address_list.defects.append(errors.InvalidHeaderDefect(
2023 2024             "invalid address in address-list"))
2024 2025         if value: # Must be a , at this point.
2025 -         address_list.append(ValueTerminal(',', 'list-separator'))
2026 +         address_list.append(ListSeparator)
2026 2027         value = value[1:]
2027 2028         return address_list, value
2028 2029

```

...est/test\_email/test\_\_header\_value\_parser.py

```

@@ -2985,6 +2985,11 @@ def
test_address_list_with_unicode_names_in_quotes(self):
2985 2985         '?utf-8?q?H=C3=BCbsch?= Kaktus <beautiful@example.com>,\n'
2986 2986         '=?utf-8?q?bei=C3=9Ft_bei=C3=9Ft?= <biter@example.com>\n')
2987 2987
2988 +     def test_address_list_with_list_separator_after_fold(self):
2989 +         to = '0123456789' * 8 + '@foo, ä <foo@bar>'
2990 +         self._test(parser.get_address_list(to)[0],
2991 +             '0123456789' * 8 + '@foo,\n =?utf-8?q?=C3=A4?=
<foo@bar>\n')
2992 +
2988 2993         # XXX Need tests with comments on various sides of a unicode token,
2989 2994         # and with unicode tokens in the comments. Spaces inside the quotes
2990 2995         # currently don't do the right thing.

```

...3-01-09-14-08-02.gh-issue-100884.DcmdLl.rst

```

@@ -0,0 +1,2 @@
1 + email: fix misfolding of comma in address-lists over multiple lines in

```

2 + combination with unicode encoding.

**Comments** 0