

NAME

dialrules – *HylaFAX* dial string processing rules

DESCRIPTION

A dial string specifies how to dial the telephone in order to reach a destination facsimile machine, or similar device. This string is supplied by a user with each outgoing facsimile job. User-supplied dial strings need to be processed in two ways by the *HylaFAX* server processes: to craft a canonical phone number for use in locating the receiver's capabilities, and to process into a form suitable for sending to a modem. In addition client applications may need to process a dial string to formulate an external form that does not include private information such as a credit card access code. Phone number canonicalization and dial string preparation are done according to *dial string processing rules* that are located in a file specified in the server configuration file; see the **DialStringRules** parameter in *hylafax-config(5F)*. The generation of an externalized form for a dial string is done by rules that optionally appear in */usr/local/lib/fax/dialrules* on client machines.

A dial string rules file is an ASCII file containing one or more *rule sets*. A rule set defines a set of *transformation rules* that are sequentially applied to a dial string. Each rule set is associated with an identifier, with certain well-known identifiers being used by the facsimile server or client application. Each transformation rule is a regular expression and a replacement string; the regular expression is repeatedly applied to a dial string and any matching substring is replaced by the replacement string.

The syntax of a dial string rules file is as follows. Comments are introduced with the “!” character and continue to the end of the current line. Identifiers are formed from a leading alphabetic and any number of subsequent alpha-numeric characters. A rule set is of the form:

```
Identifier := [
    rule1
    rule2
    ...
]
```

where *rule1*, *rule2*, and so on are transformation rules. Line breaks are significant. The initial rule set definition line and the trailing “]” must be on separate lines; and each transformation rule must also be on a single line. Transformation rules are of the form:

```
regular-expression = replacement
```

where *regular-expression* is a POSIX 1003.2 extended regular expression and *replacement* is a string that is substituted in place of any portion of the dial string that is matched by the *regular-expression*. White space is significant in parsing transformation rules. If a regular expression or replacement string has embedded white space in it, then the white space needs to be escaped with a “\” character or the entire string should be enclosed in quote (“”) marks. Replacement strings may reference the entire string matched by the regular expression with the “&” character. Substrings matched with the “(...)” constructs may be referenced by using “\n” where *n* is a single numeric digit between 1 and 9 that refers to the *n*-th matched substring; c.f. *re_format(7)*, *sed(1)*, etc.

To simplify and parameterize the construction of rule sets, dial string rules files may also include simple text-oriented variable definitions. A line of the form:

```
foo=string
```

defines a variable named *foo* that has the value *string*. String values with embedded whitespace must use the “\” character or be enclosed in quote marks. Variables are interpolated into transformation rules by referencing them as:

```
${var}
```

Note that variable interpolation is done only once, at the time a transformation rule is defined. This means that forward references are not supported and that circular definitions will not cause loops. The facsimile server automatically defines four variables to have the values defined in its configuration file: **AreaCode**, **CountryCode**, **LongDistancePrefix**, and **InternationalPrefix**. These variables are initialized before

parsing a dial string rules file; thus if they are defined in the rules file then they will override any definition by the server.

There are three well known rule set names: *CanonicalNumber* to convert a dial string to a canonical format, *DialString* to prepare a dial string before using it to dial the telephone, and *DisplayNumber* to convert a dial string to an external “displayable” form that does not include the private information that might appear in the raw dial string.

EXAMPLES

This is the default set of rules for transforming a dial string into a canonical phone number:

```

Area=${AreaCode}           ! local area code
Country=${CountryCode}     ! local country code
IDPrefix=${InternationalPrefix} ! prefix for placing an international call
LDPrefix=${LongDistancePrefix} ! prefix for placing a long distance call
!
! Convert a phone number to a canonical format:
!
! +<country><areacode><rest>
!
! by (possibly) stripping off leading dialing prefixes for
! long distance and/or international dialing.
!
CanonicalNumber := [
%.*           =           ! strip calling card stuff
[abcABC]     = 2         ! these convert alpha to numbers
[defDEF]     = 3
[ghiGHI]     = 4
[jklJKL]     = 5
[mnoMNO]     = 6
[prsPRS]     = 7
[tuvTUV]     = 8
[wxyWXY]     = 9
[^+0-9]+     =           ! strip white space etc.
^${IDPrefix} = +         ! replace int. dialing code
^${LDPrefix} = +${Country} ! replace l.d. dialing code
^[^+]       = +${Country}${Area}& ! otherwise, insert canon form
]

```

The first rule simply strips anything following a “%”; this will remove any calling card-related information. The next eight rules convert upper and lower case alphabetic to the equivalent key numbers (this is convenient for users that use mnemonic phone numbers). The tenth rule removes everything but numbers and plus signs. The eleventh rule translates any explicit international dialing prefix into the “+” symbol used to identify country codes. The twelfth rule replaces a leading long distance dialing prefix with the local country code string. The last rule matches local phone numbers and inserts the local country code and area code.

As an example, assume that

```

AreaCode=415
CountryCode=1
InternationalPrefix=011
LongDistancePrefix=1

```

then if the above set of rules is applied to “01123965-Tube%2345”, the transformations would be:

```

01123965-Tube%2345  01123965-Tube  ! strip calling card stuff
01123965-Tube       01123965-8823  ! convert alphabetic
01123965-8823       011239658823  ! strip white space etc.

```

011239658823	+239658823	! replace int. dialing code
+239658823	+239658823	! replace l.d. dialing code
+239658823	+239658823	! otherwise, insert canon form

for a final result of "+239658823".

SEE ALSO

sendfax(1), dialtest(8C), faxq(8C), faxsend(8C), faxgetty(8C), hylafax-config(5F)