NAME

SysPathSubst.pm – resolve system specific paths

SYNOPSIS

```
use SysPathSubst qw( syspathsubst compact_path);
$path = syspathsubst( $delimiter, \@components, \@templates );
$path = compact_paths( $delimiter, @paths );
```

DESCRIPTION

This package will substitute components into each element in a list of path templates.

Typically **SysPathSubst** is used on networks with multiple operating system and hardware combinations where execution and library paths differ only by a system specific path component. Using this package allows one to more easily specify those paths.

FUNCTIONS

The **SysPathSubst** module has two functions available for export:

syspathsubst

```
$path = syspathsubst( $delimiter, \@components, \@templates );
@paths = syspathsubst( undef, \@components, \@templates );
```

Perform substitution in the passed paths, and return the unique subset. Uniqueness is determined by a simple string comparison, not by resolving the paths, so that if two paths resolve to the same disk location but one uses symbolic links, they will appear distinct. To avoid this, use a path template token which ensures that links are resolved.

`@components` is an array of components to be substituted into the paths, in the order they should be checked. `@templates` is an array of templates (see "Path Templates").

If an element in `@templates` is itself an array, the path create by passing to **File::Spec::catdir** will be used in its place.

If called in a scalar environment, the subset is returned as a string of paths, with the paths separated by `$delimiter`, otherwise it is returned as a list (and `$delimiter` is ignored).

compact_paths

```
$path = compact_paths( $delimiter, @paths );
@paths = compact_paths( undef, @paths );
```

Return the unique subset of the passed paths. Uniqueness is determined by a simple string comparison, not by resolving the paths. If called in a scalar environment, the subset is returned as a string of paths, with the paths separated by `$delimiter`, otherwise it is returned as a list (and `$delimiter` is ignored).

If an element in `@paths` is itself an array, the path created by passing to **File::Spec::catdir** will be used in its place.

Path Templates

A path template is a string which can contain zero or one of a set of special tokens. The tokens indicate where in a path the system specific component will be substituted, as well as the value of two attributes, **resolvelinks** and **firstvalid**:

```
Token    resolvelinks  firstvalid
-----    -----------   ----------
 %s         false        false
 %p         false        false
 %P         false        true
 %l         true         false
 %L         true         true
```

If **resolvelinks** is true, then after substitution any symbolic links in the path are fully resolved. This always results in an absolute path.

If **firstvalid** is true, then only the first existing path resulting from the in order substitution of the system specific components is retained.

For example, if the list of system specific components is

```
c d e f
```

and the existent paths are

```
/a/c /a/e /a/f
```

Then the path template `/a/%p` would result in a list of paths

```
/a/c:/a/e:/a/f
```

while the path template `/a/%P` would result in

```
/a/c
```

COPYRIGHT AND LICENSE

Copyright 2007 The Smithsonian Astrophysical Observatory

This software is released under the GNU General Public License. You may find a copy at: <http://www.fsf.org/copyleft/gpl.html>

VERSION

This documents version 2.1.24.

Author

Diab Jerius <djerius@cfa.harvard.edu>