

Commit

✖ [3.9] gh-123270: Replaced SanitizedNames with a more surgical fix. (G... Browse files

[...H-123354](#)) ([#123432](#))

Applies changes from zipp 3.20.1 and [jaraco/zippGH-124](#)
(cherry picked from commit [2231286](#))
(cherry picked from commit [17b77bb](#))

Co-authored-by: Jason R. Coombs <jaraco@jaraco.com>

🔗 3.9 (#123432)

 jaraco committed yesterday Verified

1 parent [b4225ca](#) commit [9620552](#)

Showing 3 changed files with 87 additions and 2 deletions.

Whitesp... Ignore whitespa... Sp... Unifi...

🔍 Filter changed files

- ▼ Lib
- ▼ test
 - test_zipfile.py
 - zipfile.py
- ▼ Misc/NEWS.d/next/Library
 - 2024-08-26-13-45-2...

▼ 77 Lib/test/test_zipfile.py

3054 3054

3055 3055

3056 3056

3057 3057

3058 3058

3059 3059

3060 3060

3061 3061

3062 3062

3063 3063

3064 3064

3065 3065

3066 3066

3067 3067

3068 3068

3069 3069

3070 3070

3071 3071

data = ['/'.join(string.ascii_lowercase + str(n)) for n in range(10000)]

zipfile.CompleteDirs._implied_dirs(data)

def test_malformed_paths(self):

"""

Path should handle malformed paths gracefully.

Paths with leading slashes are not visible.

Paths with dots are treated like regular files.

"""

data = io.BytesIO()

zf = zipfile.ZipFile(data, "w")

zf.writestr("/one-slash.txt", b"content")

zf.writestr("//two-slash.txt", b"content")

zf.writestr("../parent.txt", b"content")

zf.filename = ''

root = zipfile.Path(zf)

```

3072 +         assert list(map(str, root.iterdir()))
3073 +         == ['../']
3074 +
3075 +         assert
3076 +         root.joinpath('..').joinpath('parent.txt').read
3077 +         _bytes() == b'content'
3078 +
3079 +         def test_unsupported_names(self):
3080 +             """
3081 +             Path segments with special characters
3082 +             are readable.
3083 +
3084 +             On some platforms or file systems,
3085 +             characters like
3086 +             ``:`` and ``?`` are not allowed, but
3087 +             they are valid
3088 +             in the zip file.
3089 +             """
3090 +             data = io.BytesIO()
3091 +             zf = zipfile.ZipFile(data, "w")
3092 +             zf.writestr("path?", b"content")
3093 +             zf.writestr("V: NMS.flac", b"fLaC...")
3094 +             zf.filename = ''
3095 +             root = zipfile.Path(zf)
3096 +             contents = root.iterdir()
3097 +             assert next(contents).name == 'path?'
3098 +             assert next(contents).name == 'V:
3099 +             NMS.flac'
3100 +             assert root.joinpath('V:
3101 +             NMS.flac').read_bytes() == b"fLaC..."
3102 +
3103 +         def test_backslash_not_separator(self):
3104 +             """
3105 +             In a zip file, backslashes are not
3106 +             separators.
3107 +             """
3108 +             data = io.BytesIO()
3109 +             zf = zipfile.ZipFile(data, "w")
3110 +
3111 +             zf.writestr(DirtyZipInfo.for_name("foo\\bar",
3112 +             zf), b"content")
3113 +
3114 +             zf.filename = ''
3115 +             root = zipfile.Path(zf)
3116 +             (first,) = root.iterdir()
3117 +             assert not first.is_dir()
3118 +             assert first.name == 'foo\\bar'
3119 +
3120 +             class DirtyZipInfo(zipfile.ZipInfo):
3121 +                 """
3122 +                 Bypass name sanitization.
3123 +                 """
3124 +
3125 +                 def __init__(self, filename, *args,
3126 +                 **kwargs):
3127 +                     super().__init__(filename, *args,
3128 +                     **kwargs)

```

```

3115 +         self.filename = filename
3116 +
3117 +     @classmethod
3118 +     def for_name(cls, name, archive):
3119 +         """
3120 +         Construct the same way that
3121 +         ZipFile.writestr does.
3122 +
3123 +         TODO: extract this functionality and
3124 +         re-use
3125 +         """
3126 +         self = cls(filename=name,
3127 +                     date_time=time.localtime(time.time())[:6])
3128 +         self.compress_type =
3129 +             archive.compression
3130 +         self.compress_level =
3131 +             archive.compresslevel
3132 +         if self.filename.endswith('/'): #
3133 +             pragma: no cover
3134 +             self.external_attr = 0o40775 << 16
3135 +             # drwxrwxr-x
3136 +             self.external_attr |= 0x10 # MS-
3137 +             DOS directory flag
3138 +         else:
3139 +             self.external_attr = 0o600 << 16 #
3140 +             ?rw-----
3141 +         return self
3142 +
3143 +
3144 +
3145 +
3146 +
3147 +
3148 +
3149 +
3150 +
3151 +
3152 +
3153 +
3154 +
3155 +
3156 +
3157 +
3158 +
3159 +
3160 +
3161 +
3162 +
3163 +
3164 +
3165 +
3166 +
3167 +
3168 +
3169 +
3170 +
3171 +
3172 +
3173 +
3174 +
3175 +
3176 +
3177 +
3178 +
3179 +
3180 +
3181 +
3182 +
3183 +
3184 +
3185 +
3186 +
3187 +
3188 +
3189 +
3190 +
3191 +
3192 +
3193 +
3194 +
3195 +
3196 +
3197 +
3198 +
3199 +
3200 +
3201 +
3202 +
3203 +
3204 +
3205 +
3206 +
3207 +
3208 +
3209 +
3210 +
3211 +
3212 +
3213 +
3214 +
3215 +
3216 +
3217 +
3218 +
3219 +
3220 +
3221 +
3222 +
3223 +
3224 +
3225 +
3226 +
3227 +
3228 +
3229 +
3230 +
3231 +
3232 +
3233 +
3234 +
3235 +
3236 +
3237 +
3238 +
3239 +
3240 +
3241 +
3242 +
3243 +
3244 +
3245 +
3246 +
3247 +
3248 +
3249 +
3250 +
3251 +
3252 +
3253 +
3254 +
3255 +
3256 +
3257 +
3258 +
3259 +
3260 +
3261 +
3262 +
3263 +
3264 +
3265 +
3266 +
3267 +
3268 +
3269 +
3270 +
3271 +
3272 +
3273 +
3274 +
3275 +
3276 +
3277 +
3278 +
3279 +
3280 +
3281 +
3282 +
3283 +
3284 +
3285 +
3286 +
3287 +
3288 +
3289 +
3290 +
3291 +
3292 +
3293 +
3294 +
3295 +
3296 +
3297 +
3298 +
3299 +
3300 +
3301 +
3302 +
3303 +
3304 +
3305 +
3306 +
3307 +
3308 +
3309 +
3310 +
3311 +
3312 +
3313 +
3314 +
3315 +
3316 +
3317 +
3318 +
3319 +
3320 +
3321 +
3322 +
3323 +
3324 +
3325 +
3326 +
3327 +
3328 +
3329 +
3330 +
3331 +
3332 +
3333 +
3334 +
3335 +
3336 +
3337 +
3338 +
3339 +
3340 +
3341 +
3342 +
3343 +
3344 +
3345 +
3346 +
3347 +
3348 +
3349 +
3350 +
3351 +
3352 +
3353 +
3354 +
3355 +
3356 +
3357 +
3358 +
3359 +
3360 +
3361 +
3362 +
3363 +
3364 +
3365 +
3366 +
3367 +
3368 +
3369 +
3370 +
3371 +
3372 +
3373 +
3374 +
3375 +
3376 +
3377 +
3378 +
3379 +
3380 +
3381 +
3382 +
3383 +
3384 +
3385 +
3386 +
3387 +
3388 +
3389 +
3390 +
3391 +
3392 +
3393 +
3394 +
3395 +
3396 +
3397 +
3398 +
3399 +
3400 +
3401 +
3402 +
3403 +
3404 +
3405 +
3406 +
3407 +
3408 +
3409 +
3410 +
3411 +
3412 +
3413 +
3414 +
3415 +
3416 +
3417 +
3418 +
3419 +
3420 +
3421 +
3422 +
3423 +
3424 +
3425 +
3426 +
3427 +
3428 +
3429 +
3430 +
3431 +
3432 +
3433 +
3434 +
3435 +
3436 +
3437 +
3438 +
3439 +
3440 +
3441 +
3442 +
3443 +
3444 +
3445 +
3446 +
3447 +
3448 +
3449 +
3450 +
3451 +
3452 +
3453 +
3454 +
3455 +
3456 +
3457 +
3458 +
3459 +
3460 +
3461 +
3462 +
3463 +
3464 +
3465 +
3466 +
3467 +
3468 +
3469 +
3470 +
3471 +
3472 +
3473 +
3474 +
3475 +
3476 +
3477 +
3478 +
3479 +
3480 +
3481 +
3482 +
3483 +
3484 +
3485 +
3486 +
3487 +
3488 +
3489 +
3490 +
3491 +
3492 +
3493 +
3494 +
3495 +
3496 +
3497 +
3498 +
3499 +
3500 +
3501 +
3502 +
3503 +
3504 +
3505 +
3506 +
3507 +
3508 +
3509 +
3510 +
3511 +
3512 +
3513 +
3514 +
3515 +
3516 +
3517 +
3518 +
3519 +
3520 +
3521 +
3522 +
3523 +
3524 +
3525 +
3526 +
3527 +
3528 +
3529 +
3530 +
3531 +
3532 +
3533 +
3534 +
3535 +
3536 +
3537 +
3538 +
3539 +
3540 +
3541 +
3542 +
3543 +
3544 +
3545 +
3546 +
3547 +
3548 +
3549 +
3550 +
3551 +
3552 +
3553 +
3554 +
3555 +
3556 +
3557 +
3558 +
3559 +
3560 +
3561 +
3562 +
3563 +
3564 +
3565 +
3566 +
3567 +
3568 +
3569 +
3570 +
3571 +
3572 +
3573 +
3574 +
3575 +
3576 +
3577 +
3578 +
3579 +
3580 +
3581 +
3582 +
3583 +
3584 +
3585 +
3586 +
3587 +
3588 +
3589 +
3590 +
3591 +
3592 +
3593 +
3594 +
3595 +
3596 +
3597 +
3598 +
3599 +
3600 +
3601 +
3602 +
3603 +
3604 +
3605 +
3606 +
3607 +
3608 +
3609 +
3610 +
3611 +
3612 +
3613 +
3614 +
3615 +
3616 +
3617 +
3618 +
3619 +
3620 +
3621 +
3622 +
3623 +
3624 +
3625 +
3626 +
3627 +
3628 +
3629 +
3630 +
3631 +
3632 +
3633 +
3634 +
3635 +
3636 +
3637 +
3638 +
3639 +
3640 +
3641 +
3642 +
3643 +
3644 +
3645 +
3646 +
3647 +
3648 +
3649 +
3650 +
3651 +
3652 +
3653 +
3654 +
3655 +
3656 +
3657 +
3658 +
3659 +
3660 +
3661 +
3662 +
3663 +
3664 +
3665 +
3666 +
3667 +
3668 +
3669 +
3670 +
3671 +
3672 +
3673 +
3674 +
3675 +
3676 +
3677 +
3678 +
3679 +
3680 +
3681 +
3682 +
3683 +
3684 +
3685 +
3686 +
3687 +
3688 +
3689 +
3690 +
3691 +
3692 +
3693 +
3694 +
3695 +
3696 +
3697 +
3698 +
3699 +
3700 +
3701 +
3702 +
3703 +
3704 +
3705 +
3706 +
3707 +
3708 +
3709 +
3710 +
3711 +
3712 +
3713 +
3714 +
3715 +
3716 +
3717 +
3718 +
3719 +
3720 +
3721 +
3722 +
3723 +
3724 +
3725 +
3726 +
3727 +
3728 +
3729 +
3730 +
3731 +
3732 +
3733 +
3734 +
3735 +
3736 +
3737 +
3738 +
3739 +
3740 +
3741 +
3742 +
3743 +
3744 +
3745 +
3746 +
3747 +
3748 +
3749 +
3750 +
3751 +
3752 +
3753 +
3754 +
3755 +
3756 +
3757 +
3758 +
3759 +
3760 +
3761 +
3762 +
3763 +
3764 +
3765 +
3766 +
3767 +
3768 +
```

9      Lib/zipfile.py

```

2146 2146 def _ancestry(path):
2147 2147     """
2148 2148     Given a path with elements separated by
2149 2149     -   posixpath.sep, generate all elements of
        that path
        2149 2149     +   posixpath.sep, generate all elements of
        that path.
2150 2150
2151 2151     >>> list(_ancestry('b/d'))
2152 2152     ['b/d', 'b']
2158 2158     ['b']
2159 2159     >>> list(_ancestry(''))
2160 2160     []
        2161 2161     +
        2162 2162     +   Multiple separators are treated like a
        single.
        2163 2163     +
        2164 2164     +   >>> list(_ancestry('//b//d///f//'))
        2165 2165     +   ['//b//d///f', '//b//d', '//b']
2161 2166     """
2162 2167     path = path.rstrip(posixpath.sep)
2163 2167     -   while path and path != posixpath.sep:
        2168 2167     +   while path.rstrip(posixpath.sep):

```

2164	2169	<code>yield</code> path
2165	2170	path, tail = posixpath. <code>split</code> (path)
2166	2171	

▼ 3 ■■■■■

Misc/NEWS.d/next/Library/2024-08-26-13-45-20.gh-issue-1232... 

...	...	<code>@@ -0,0 +1,3 @@</code>
	1	+ Applied a more surgical fix for malformed payloads in <code>:class:`zipfile.Path`</code>
	2	+ causing infinite loops (gh-122905) without breaking contents using
	3	+ legitimate characters.

0 comments on commit 9620552

Please [sign in](#) to comment.