It seems like there should have been a compatibility call left in place for the removal of URI escape between stdlib 2.x and 3.x

As seen here:
https://github.com/qoobaa/s3/issues/132

Various projects are breaking due to the move of escape to DEFAULT_PARSER

```
--- /home/kallisti5/.gem/ruby/3.0.0/gems/s3-0.3.29/lib/s3/bucket.rb.original 2022-02-18 13:26:37.247078560 -0600
+++ /home/kallisti5/.gem/ruby/3.0.0/gems/s3-0.3.29/lib/s3/bucket.rb 2022-02-18 13:26:47.707146732 -0600
@@ -151,7 +151,7 @@
  # If there are more than 1000 objects S3 truncates listing and
  # we need to request another listing for the remaining objects.
  while parse_is_truncated(response.body)
-     next_request_options = {:marker => URI.escape(objects_attributes.last[:key])}
+     next_request_options = {:marker => URI::DEFAULT_PARSER.escape(objects_attributes.last[:key])}
  if max_keys
     break if objects_attributes.length >= max_keys

Related issues:
Related to Ruby master - Misc #17309: URI.escape being deprecated, yet there ...
```

History
---

#1 - 02/18/2022 07:45 PM - jeremyevans0 (Jeremy Evans)
- Backport deleted (2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN)
- ruby -v deleted (3.0.3p157)
- Subject changed from URI escape compatibility function? to Add back URI.escape
- Tracker changed from Bug to Feature

URI.escape has been deprecated since Ruby 1.9. Unfortunately, the deprecation warnings were only emitted in verbose warning mode until Ruby 2.6. However, Ruby 2.7 included the deprecation warning even in non-verbose warning mode. Ruby 3.0 removed the method. So this is definitely not a bug, it is expected behavior. Switching to feature, though I would guess there is a very low chance of us reintroducing the method.

#2 - 02/19/2022 07:55 AM - byroot (Jean Boussier)
- Related to Misc #17309: URI.escape being deprecated, yet there is no replacement added