

# Future direction of energy storage power stations

Source: <https://angulate.co.za/Thu-28-Jan-2021-17548.html>

Website: <https://angulate.co.za>

This PDF is generated from: <https://angulate.co.za/Thu-28-Jan-2021-17548.html>

Title: Future direction of energy storage power stations

Generated on: 2026-02-11 17:54:03

Copyright (C) 2026 ANGULATE CONTAINERS. All rights reserved.

For the latest updates and more information, visit our website: <https://angulate.co.za>

---

The get member function waits (by calling `wait()`) until the shared state is ready, then retrieves the value stored in the shared state (if any). Right after calling this function, valid ...

The promise is the "push" end of the promise-future communication channel: the operation that stores a value in the shared state synchronizes-with (as defined in ...)

Specifies state of a future as returned by `wait_for` and `wait_until` functions of `std::future` and `std::shared_future`. Constants

Checks if the future refers to a shared state. This is the case only for futures that were not default-constructed or moved from (i.e. returned by `std::promise::get_future()`, ...)

If the future is the result of a call to `std::async` that used lazy evaluation, this function returns immediately without waiting. This function may block for longer than ...

The class template `std::future` provides a mechanism to access the result of asynchronous operations: An asynchronous operation (created via `std::async`, ...)

In summary: `std::future` is an object used in multithreaded programming to receive data or an exception from a different thread; it is one end of a single-use, one-way ...

A future represents the result of an asynchronous operation, and can have two states: uncompleted or completed. Most likely, as you aren't doing this just for fun, you actually ...

If the future is the result of a call to `async` that used lazy evaluation, this function returns immediately without

# Future direction of energy storage power stations

Source: <https://angulate.co.za/Thu-28-Jan-2021-17548.html>

Website: <https://angulate.co.za>

waiting. The behavior is undefined if valid () is false before the call ...

```
future (const future & ) = delete; ~future (); future & operator =(const future & ) = delete; future & operator
=(future & & ) noexcept; shared_future <R> share () noexcept; // ...
```

Web: <https://angulate.co.za>

