

Future home solar power generation system



[Ansible yum throwing future feature annotations is not defined](#)

The error: SyntaxError: future feature annotations is not defined usually related to an old version of python, but my remote server has Python3.9 and to verify it - I also added it in my

[Complete Guide To Fully Solar Powered Houses \(2025\) , Costs](#)

Everything you need to know about fully solar powered houses in 2025. Complete cost analysis, installation guide, real examples, and expert insights. Start your solar journey today.



std::future

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

std::future::valid

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()`),



[Solar Power For Home In 2025: What's Changed.](#)



This article explores the latest changes in solar technology, cost reductions, and smart innovations that make solar power for home more

[Home Solar Energy Solutions , Hybrid & Off Grid Solar](#)

Discover reliable solar energy systems at SunGoldPower. We specialize in off-grid, hybrid, and commercial solar power systems designed to maximize energy



[The Future is Solar: Revolutionizing Residential Solar](#)

With the average home solar system size expanding as well, solar technology is maturing into a mainstream energy source. This article will explore

std::future::wait_until

wait_until waits for a result to become available. It blocks until specified timeout_time has been reached or the result becomes available, whichever comes first. The return value indicates why



std::future::get

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 () is false.

std::future::wait_for

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 `timeout_duration` due to



[Solar Powered Whole House Generator Guide for 2026](#)

Choosing the ideal solar powered whole house generator involves careful planning and evaluation. This step-by-step guide will walk you through

[12 Best Solar Generators for Home Backup in 2026 \(Tested & Reviewed\)](#)

Discover the 12 best solar generators for home backup in 2026, tested, compared, and reviewed for real emergency use.



[The best solar generators for 2026, tested and](#)

Whether you're camping, living off-grid, or want backup power for your house, the best solar generators help you tap into

[The Future of Residential Solar Energy: What's Next for](#)

In this article, we'll explore what defines a modern residential



std::future_status



std::future::~~future

Releases any shared state. This means: If the current object holds the last reference to its shared state, the shared state is destroyed. The current object gives up its reference to its shared

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



Standard library header (C++11)

```
future (const future &) = delete; ~future ();  
future & operator =(const future &) = delete;  
future & operator =(future &&) noexcept;  
shared_future share () noexcept; // retrieving the  
value
```

std::shared_future

Unlike `std::future`, which is only moveable (so only one instance can refer to any particular asynchronous result), `std::shared_future` is copyable and multiple shared future objects



Contact Us

For off-grid system quotes, technical support, or partnerships, please visit:
<https://kephamatraining.co.za>