ns3::ApplicationContainer Class Reference

holds a vector of ns3::Application pointers. More...

```
#include "application-container.h"
```

▸ Collaboration diagram for ns3::ApplicationContainer:

Public Types

typedef std::vector< Ptr< Application > >::const_iterator	Iterator Application container iterator. More...

Public Member Functions

	ApplicationContainer () Create an empty **ApplicationContainer**. More...
	ApplicationContainer (Ptr< Application > application) Create an **ApplicationContainer** with exactly one application which has been previously instantiated. More...
	ApplicationContainer (std::string name) Create an **ApplicationContainer** with exactly one application which has been previously instantiated and assigned a name using the Object Name Service. More...
void	Add (ApplicationContainer other) Append the contents of another **ApplicationContainer** to the end of this container. More...
void	Add (Ptr< Application > application) Append a single Ptr<Application> to this container. More...
void	Add (std::string name) Append to this container the single Ptr<Application> referred to via its object name service registered name. More...
Iterator	Begin (void) const Get an iterator which refers to the first Application in the container. More...
Iterator	End (void) const Get an iterator which indicates past-the-last Application in the container. More...
Ptr< Application >	Get (uint32_t i) const Get the Ptr<Application> stored in this container at a given index. More...
uint32_t	GetN (void) const Get the number of Ptr<Application> stored in this container. More...
void	Start (Time start) Arrange for all of the Applications in this container to Start() at the Time given as a parameter. More...
void	StartWithJitter (Time start, Ptr< RandomVariableStream > rv) Start all of the Applications in this container at the start time given as a parameter, plus some jitter. More...
void	Stop (Time stop) Arrange for all of the Applications in this container to Stop() at the Time given as a parameter. More...

Private Attributes

std::vector< Ptr< Application > >	m_applications Applications smart pointers. More...

Detailed Description

holds a vector of ns3::Application pointers.

Typically ns-3 Applications are installed on nodes using an Application helper. The helper Install method takes a NodeContainer which holds some number of Ptr<Node>. For each of the Nodes in the NodeContainer the helper will instantiate an application, install it in a node and add a Ptr<Application> to that application into a Container for use by the caller. This is that container used to hold the Ptr<Application> which are instantiated by the Application helper.

Definition at line 42 of file application-container.h.

Member Typedef Documentation

◆ Iterator

typedef std::vector<Ptr<Application> >::const_iterator ns3::ApplicationContainer::Iterator

Application container iterator.

Definition at line 69 of file application-container.h.

Constructor & Destructor Documentation

◆ ApplicationContainer() `[1/3]`

ns3::ApplicationContainer::ApplicationContainer ( )

Create an empty ApplicationContainer.

Definition at line 29 of file application-container.cc.

◆ ApplicationContainer() `[2/3]`

ns3::ApplicationContainer::ApplicationContainer (Ptr< Application > application)

Create an ApplicationContainer with exactly one application which has been previously instantiated.

The single application is specified by a smart pointer.

Parameters

application The Ptr<Application> to add to the container.

Definition at line 33 of file application-container.cc.

References m_applications.

◆ ApplicationContainer() `[3/3]`

ns3::ApplicationContainer::ApplicationContainer (std::string name)

Create an ApplicationContainer with exactly one application which has been previously instantiated and assigned a name using the Object Name Service.

This Application is then specified by its assigned name.

Parameters

name The name of the Application Object to add to the container.

Definition at line 38 of file application-container.cc.

References m_applications.

Member Function Documentation

◆ Add() `[1/3]`

void ns3::ApplicationContainer::Add (ApplicationContainer other)

Append the contents of another ApplicationContainer to the end of this container.

Parameters

other The ApplicationContainer to append.

Definition at line 67 of file application-container.cc.

References Begin(), End(), and m_applications.

Referenced by BuildAppsTest(), CsmaBroadcastTestCase::DoRun(), EpcS1uUlTestCase::DoRun(), CsmaStarTestCase::DoRun(), experiment(), LteAggregationThroughputScaleTestCase::GetThroughput(), ns3::V4PingHelper::Install(), ns3::PacketSinkHelper::Install(), ns3::ThreeGppHttpClientHelper::Install(), ns3::WaveBsmHelper::Install(), ns3::Ping6Helper::Install(), ns3::UdpServerHelper::Install(), ns3::BulkSendHelper::Install(), ns3::OnOffHelper::Install(), ns3::UdpEchoServerHelper::Install(), ns3::RadvdHelper::Install(), ns3::ThreeGppHttpServerHelper::Install(), ns3::UdpClientHelper::Install(), ns3::UdpTraceClientHelper::Install(), ns3::UdpEchoClientHelper::Install(), and ns3::DhcpHelper::InstallDhcpClient().

▸ Here is the call graph for this function:

▸ Here is the caller graph for this function:

◆ Add() [2/3]

void ns3::ApplicationContainer::Add (Ptr< Application > application)

Append a single Ptr<Application> to this container.

Parameters

application The Ptr<Application> to append.

Definition at line 75 of file application-container.cc.

References m_applications.

◆ Add() [3/3]

void ns3::ApplicationContainer::Add (std::string name)

Append to this container the single Ptr<Application> referred to via its object name service registered name.

Parameters

name The name of the Application Object to add to the container.

Definition at line 80 of file application-container.cc.

References m_applications.

◆ Begin()

ApplicationContainer::Iterator ns3::ApplicationContainer::Begin (void) const

Get an iterator which refers to the first Application in the container.

Applications can be retrieved from the container in two ways. First, directly by an index into the container, and second, using an iterator. This method is used in the iterator method and is typically used in a for-loop to run through the Applications

```
ApplicationContainer::Iterator i;
for (i = container.Begin (); i != container.End (); ++i)
  {
    (*i)->method ();  // some Application method
  }
```

Returns

an iterator which refers to the first Application in the container.

Definition at line 46 of file application-container.cc.

References m_applications.

Referenced by Add(), ns3::WaveBsmHelper::Install(), Start(), StartWithJitter(), and Stop().

Here is the caller graph for this function:

◆ End()

ApplicationContainer::Iterator ns3::ApplicationContainer::End (void) const

Get an iterator which indicates past-the-last Application in the container.

Applications can be retrieved from the container in two ways. First, directly by an index into the container, and second, using an iterator. This method is used in the iterator method and is typically used in a for-loop to run through the Applications

```
ApplicationContainer::Iterator i;
for (i = container.Begin (); i != container.End (); ++i)
  {
    (*i)->method ();  // some Application method
  }
```

Returns

an iterator which indicates an ending condition for a loop.

Definition at line 51 of file application-container.cc.

References m_applications.

Referenced by Add(), ns3::WaveBsmHelper::Install(), Start(), StartWithJitter(), and Stop().

Here is the caller graph for this function:

◆ Get()

Ptr< Application > ns3::ApplicationContainer::Get (uint32_t i) const

Get the Ptr<Application> stored in this container at a given index.

Applications can be retrieved from the container in two ways. First, directly by an index into the container, and second, using an iterator. This method is used in the direct method and is used to retrieve the indexed Ptr<Appliation>.

```
uint32_t nApplications = container.GetN ();
for (uint32_t i = 0 i < nApplications; ++i)
  {
    Ptr<Application> p = container.Get (i)
    i->method ();  // some Application method
  }
```

Parameters

i the index of the requested application pointer.

Returns

the requested application pointer.

Definition at line 62 of file application-container.cc.

References m_applications.

Referenced by WifiMsduAggregatorThroughputTest::DoRun(), WifiAcMappingTest::DoRun(), DhcpTestCase::DoRun(), LteX2HandoverTestCase::DoRun(), BriteTopologyFunctionTestCase::DoRun(), EpcS1uDlTestCase::DoRun(), ThreeGppHttpObjectTestCase::DoRun(), LteX2HandoverMeasuresTestCase::DoRun(), LteEpcE2eDataTestCase::DoRun(), EpcS1uUlTestCase::DoRun(), GoodputSampling(), and Experiment::Run().

Here is the caller graph for this function:

◆ GetN()

uint32_t ns3::ApplicationContainer::GetN (void) const

Get the number of **Ptr<Application>** stored in this container.

Applications can be retrieved from the container in two ways. First, directly by an index into the container, and second, using an iterator. This method is used in the direct method and is typically used to define an ending condition in a for-loop that runs through the stored Applications

```
uint32_t nApplications = container.GetN ();
for (uint32_t i = 0 i < nApplications; ++i)
  {
    Ptr<Application> p = container.Get (i)
    i->method ();  // some Application method
  }
```

Returns

the number of **Ptr<Application>** stored in this container.

Definition at line **57** of file **application-container.cc**.

References **m_applications**.

Referenced by **ThreeGppHttpObjectTestCase::DoRun()**.

Here is the caller graph for this function:

◆ Start()

void ns3::ApplicationContainer::Start (Time start)

Arrange for all of the Applications in this container to Start() at the Time given as a parameter.

All Applications need to be provided with a starting simulation time and a stopping simulation time. The ApplicationContainer is a convenient place for allowing all of the contained Applications to be told to wake up and start doing their thing (Start) at a common time.

This method simply iterates through the contained Applications and calls their Start() methods with the provided Time.

Parameters

start The Time at which each of the applications should start.

Definition at line 87 of file application-container.cc.

References Begin(), End(), ns3::Application::SetStartTime(), and visualizer.core::start().

Referenced by Experiment::ApplicationSetup(), BuildAppsTest(), CreateBulkFlow(), CreateOnOffFlow(), WifiMsduAggregatorThroughputTest::DoRun(), WifiAcMappingTest::DoRun(), Ns3TcpNoDelayTestCase::DoRun(), Ns3TcpSocketTestCase1::DoRun(), NscTcpLossTestCase1::DoRun(), UdpClientServerTestCase::DoRun(), DhcpTestCase::DoRun(), CsmaBridgeTestCase::DoRun(), Ns3TcpLossTestCase::DoRun(), Ns3TcpStateTestCase::DoRun(), BriteTopologyFunctionTestCase::DoRun(), Ns3TcpInteroperabilityTestCase::DoRun(), LteIpv6RoutingTestCase::DoRun(), EpcS1uDlTestCase::DoRun(), LteEpcE2eDataTestCase::DoRun(), UdpTraceClientServerTestCase::DoRun(), Ns3TcpSocketTestCase2::DoRun(), NscTcpLossTestCase2::DoRun(), CsmaBroadcastTestCase::DoRun(), Ns3TcpCwndTestCase1::DoRun(), CsmaMulticastTestCase::DoRun(), EpcS1uUlTestCase::DoRun(), Ns3TcpCwndTestCase2::DoRun(), CsmaOneSubnetTestCase::DoRun(), CsmaPacketSocketTestCase::DoRun(), CsmaPingTestCase::DoRun(), CsmaRawIpSocketTestCase::DoRun(), CsmaStarTestCase::DoRun(), LteAggregationThroughputScaleTestCase::GetThroughput(), ns3::WaveBsmHelper::Install(), AodvExample::InstallApplications(), DsdvManetExample::InstallApplications(), NetAnimExperiment::Run(), RoutingExperiment::Run(), Experiment::Run(), and RoutingHelper::SetupRoutingMessages().

Here is the call graph for this function:

Here is the caller graph for this function:

◆ StartWithJitter()

```
void ns3::ApplicationContainer::StartWithJitter ( Time                                start,
                                                  Ptr< RandomVariableStream >         rv
                                                )
```

Start all of the Applications in this container at the start time given as a parameter, plus some jitter.

This method iterates through the contained Applications and calls their Start() methods with the provided start Time, plus a jitter value drawn from the provided random variable.

Parameters

start	The Time at which each of the applications should start.
rv	The random variable that adds jitter (units of seconds)

Definition at line 96 of file application-container.cc.

References Begin(), End(), ns3::RandomVariableStream::GetValue(), NS_LOG_DEBUG, ns3::Seconds(), ns3::Application::SetStartTime(), and visualizer.core::start().

▸ Here is the call graph for this function:

◆ Stop()

void ns3::ApplicationContainer::Stop (Time stop)

Arrange for all of the Applications in this container to Stop() at the Time given as a parameter.

All Applications need to be provided with a starting simulation time and a stopping simulation time. The ApplicationContainer is a convenient place for allowing all of the contained Applications to be told to shut down and stop doing their thing (Stop) at a common time.

This method simply iterates through the contained Applications and calls their Stop() methods with the provided Time.

Parameters

stop	The Time at which each of the applications should stop.

Definition at line 107 of file application-container.cc.

References Begin(), End(), and ns3::Application::SetStopTime().

Referenced by Experiment::ApplicationSetup(), BuildAppsTest(), CreateBulkFlow(), CreateOnOffFlow(), WifiMsduAggregatorThroughputTest::DoRun(), WifiAcMappingTest::DoRun(), Ns3TcpNoDelayTestCase::DoRun(), Ns3TcpSocketTestCase1::DoRun(), NscTcpLossTestCase1::DoRun(), DhcpTestCase::DoRun(), UdpClientServerTestCase::DoRun(), CsmaBridgeTestCase::DoRun(), Ns3TcpLossTestCase::DoRun(), Ns3TcpStateTestCase::DoRun(), BriteTopologyFunctionTestCase::DoRun(), Ns3TcpInteroperabilityTestCase::DoRun(), LteIpv6RoutingTestCase::DoRun(), EpcS1uDlTestCase::DoRun(), UdpTraceClientServerTestCase::DoRun(), Ns3TcpSocketTestCase2::DoRun(), CsmaBroadcastTestCase::DoRun(), NscTcpLossTestCase2::DoRun(), Ns3TcpCwndTestCase1::DoRun(), CsmaMulticastTestCase::DoRun(), EpcS1uUlTestCase::DoRun(), Ns3TcpCwndTestCase2::DoRun(), CsmaOneSubnetTestCase::DoRun(), CsmaPacketSocketTestCase::DoRun(), CsmaPingTestCase::DoRun(), CsmaRawIpSocketTestCase::DoRun(), CsmaStarTestCase::DoRun(), ns3::WaveBsmHelper::Install(), AodvExample::InstallApplications(), DsdvManetExample::InstallApplications(), NetAnimExperiment::Run(), Experiment::Run(), and RoutingHelper::SetupRoutingMessages().

Here is the call graph for this function:

Here is the caller graph for this function:

Member Data Documentation

◆ m_applications

std::vector<Ptr<Application> > ns3::ApplicationContainer::m_applications private

Applications smart pointers.

Definition at line 227 of file application-container.h.

Referenced by Add(), ApplicationContainer(), Begin(), End(), Get(), and GetN().

The documentation for this class was generated from the following files:

- src/network/helper/**application-container.h**
- src/network/helper/**application-container.cc**