

# TraDemGen Reference Manual

## 1.00.2

Generated by Doxygen 1.4.7

Sat Jun 27 22:19:46 2015

## Contents

<b>1</b>	<b>TraDemGen Documentation</b>	<b>1</b>
<b>2</b>	<b>TraDemGen Directory Hierarchy</b>	<b>2</b>
<b>3</b>	<b>TraDemGen Namespace Index</b>	<b>3</b>
<b>4</b>	<b>TraDemGen Hierarchical Index</b>	<b>3</b>
<b>5</b>	<b>TraDemGen Class Index</b>	<b>7</b>
<b>6</b>	<b>TraDemGen File Index</b>	<b>10</b>
<b>7</b>	<b>TraDemGen Page Index</b>	<b>12</b>
<b>8</b>	<b>TraDemGen Directory Documentation</b>	<b>13</b>
<b>9</b>	<b>TraDemGen Namespace Documentation</b>	<b>15</b>
<b>10</b>	<b>TraDemGen Class Documentation</b>	<b>30</b>
<b>11</b>	<b>TraDemGen File Documentation</b>	<b>167</b>
<b>12</b>	<b>TraDemGen Page Documentation</b>	<b>198</b>

## 1 TraDemGen Documentation

### 1.1 Getting Started

- [Main features](#)
- [Installation](#)
- [Linking with TraDemGen](#)
- [Users Guide](#)
- [Tutorials](#)
- [Copyright and License](#)
- [Make a Difference](#)
- [Make a new release](#)
- [People](#)

## 1.2 TraDemGen at SourceForge

- [Project page](#)
- [Download TraDemGen](#)
- [Open a ticket for a bug or feature](#)
- [Mailing lists](#)
- [Forums](#)
  - [Discuss about Development issues](#)
  - [Ask for Help](#)
  - [Discuss TraDemGen](#)

## 1.3 TraDemGen Development

- [Git Repository](#) (Subversion is deprecated)
- [Coding Rules](#)
- [Documentation Rules](#)
- [Test Rules](#)

## 1.4 External Libraries

- [Boost](#) (C++ STL extensions)
- [Python](#)
- [MySQL client](#)
- [SOI](#) (C++ DB API)

## 1.5 Support TraDemGen

## 1.6 About TraDemGen

TraDemGen aims at providing a clean API, and the corresponding C++ implementation, able to generate demand for travel solutions (e.g., from JFK to PEK on 25-05-2009) according to characteristics (e.g., Willingness-To-Pay, preferred airline, etc). TraDemGen mainly targets simulation purposes. [N](#)

TraDemGen makes an extensive use of existing open-source libraries for increased functionality, speed and accuracy. In particular the [Boost](#) (C++ *Standard Extensions*) library is used.

The TraDemGen library originates from the department of Operational Research and Innovation at [Amadeus](#), Sophia Antipolis, France. TraDemGen is released under the terms of the [GNU Lesser General Public License](#) (LGPLv2.1) for you to enjoy.

TraDemGen should work on [GNU/Linux](#), [Sun Solaris](#), Microsoft Windows (with [Cygwin](#), [MinGW/MSYS](#), or [Microsoft Visual C++ .NET](#)) and [Mac OS X](#) operating systems.

**Note:**

(N) - The TraDemGen library is **NOT** intended, in any way, to be used by airlines for production systems. If you want to report issue, bug or feature request, or if you just want to give feedback, have a look on the right-hand side of this page for the preferred reporting methods. In any case, please do not contact Amadeus directly for any matter related to TraDemGen.

## 2 TraDemGen Directory Hierarchy

### 2.1 TraDemGen Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

<b>test</b>	<b>14</b>
<b>trademgen</b>	<b>15</b>
<b>trademgen</b>	<b>15</b>
<b>basic</b>	<b>13</b>
<b>batches</b>	<b>13</b>
<b>bom</b>	<b>13</b>
<b>command</b>	<b>14</b>
<b>factory</b>	<b>14</b>
<b>python</b>	<b>14</b>
<b>service</b>	<b>14</b>
<b>ui</b>	<b>15</b>
<b>cmdline</b>	<b>14</b>

## 3 TraDemGen Namespace Index

### 3.1 TraDemGen Namespace List

Here is a list of all namespaces with brief descriptions:

<b>SEVMGR</b>	<b>15</b>
<b>stdair</b> (Forward declarations )	<b>16</b>
<b>TRADEMGEN</b>	<b>16</b>
<b>TRADEMGEN::DemandParserHelper</b>	<b>26</b>

## 4 TraDemGen Hierarchical Index

### 4.1 TraDemGen Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

std::allocator< T >	
std::auto_ptr< T >	
std::basic_string< Char >	
std::basic_string< char >	
std::string	
std::basic_string< wchar_t >	
std::wstring	
std::bitset< Bits >	
<b>BomAbstract</b>	<b>30</b>
TRADEMGEN::DemandStream	68
TRADEMGEN::BomDisplay	30
grammar	31
TRADEMGEN::DemandParserHelper::DemandParser	57
stdair::CategoricalAttribute< T >	31
TRADEMGEN::CategoricalAttributeLite< T >	33
<b>CmdAbstract</b>	<b>36</b>
TRADEMGEN::DemandFileParser	52
TRADEMGEN::DemandManager	55
TRADEMGEN::DemandParser	56
std::complex	
TRADEMGEN::ContinuousAttribute< T >	36
TRADEMGEN::ContinuousAttributeLite< T >	38
<b>TestFixture</b>	<b>40</b>
DemandGenerationTestSuite	54
TRADEMGEN::DBManager	40
TRADEMGEN::DefaultMap	45
TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >	59
std::deque< T >	
<b>TRADEMGEN::DictionaryManager</b>	<b>91</b>
std::exception	
std::bad_alloc	
std::bad_cast	

std::bad_exception	
std::bad_typeid	
std::ios_base::failure	
std::logic_error	
std::domain_error	
std::invalid_argument	
std::length_error	
std::out_of_range	
std::runtime_error	
std::overflow_error	
std::range_error	
std::underflow_error	
<b>FacServiceAbstract</b>	<b>94</b>
<b>TRADEMGEN::FacTRADEMGENServiceContext</b>	<b>94</b>
<b>FileNotFoundException</b>	<b>96</b>
<b>TRADEMGEN::DemandInputFileNotFoundException</b>	<b>55</b>
<b>TRADEMGEN::FlagSaver</b>	<b>96</b>
<b>InputFilePath</b>	<b>98</b>
<b>TRADEMGEN::DemandFilePath</b>	<b>53</b>
std::ios_base	
std::basic_ios	
std::basic_istream	
std::basic_ifstream	
std::basic_iostream	
std::basic_fstream	
std::basic_stringstream	
std::basic_istringstream	
std::basic_ostream	
std::basic_iostream	
std::basic_ofstream	
std::basic_ostringstream	
std::basic_ios< char >	
std::basic_istream< char >	
std::basic_ifstream< char >	
std::ifstream	
std::basic_iostream< char >	
std::basic_fstream< char >	
std::fstream	
std::basic_stringstream< char >	
std::stringstream	
std::basic_istringstream< char >	
std::istringstream	
std::istream	
std::basic_ostream< char >	
std::basic_iostream< char >	
std::basic_ofstream< char >	
std::ofstream	
std::basic_ostringstream< char >	

```

        std::ostringstream
    std::ostream
std::ios
std::basic_ios< wchar_t >
    std::basic_istream< wchar_t >
        std::basic_ifstream< wchar_t >
            std::wifstream
        std::basic_iostream< wchar_t >
            std::basic_fstream< wchar_t >
                std::wfstream
            std::basic_stringstream< wchar_t >
                std::wstringstream
    std::basic_istreamstream< wchar_t >
        std::wistreamstream
    std::wistream
std::basic_ostream< wchar_t >
    std::basic_iostream< wchar_t >
        std::basic_ofstream< wchar_t >
            std::wofstream
        std::basic_ostringstream< wchar_t >
            std::wostringstream
    std::wostream
std::wios

```

**KeyAbstract** 98

**TRADEMGEN::DemandStreamKey** 82

```

std::list< T >
std::map< K, T >
std::multimap< K, T >
std::multiset< K >

```

**TRADEMGEN::DemandParserHelper::ParserSemanticAction** 98

**TRADEMGEN::DemandParserHelper::doEndDemand** 92

**TRADEMGEN::DemandParserHelper::storeChannelCode** 103

**TRADEMGEN::DemandParserHelper::storeChannelProbMass** 105

**TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility** 106

**TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb** 108

**TRADEMGEN::DemandParserHelper::storeDemandMean** 110

**TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility** 111

**TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb** 113

**TRADEMGEN::DemandParserHelper::storeDemandStdDev** 115

**TRADEMGEN::DemandParserHelper::storeDestination** 116

**TRADEMGEN::DemandParserHelper::storeDow** 118

TRADEMGEN::DemandParserHelper::storeDTD	119
TRADEMGEN::DemandParserHelper::storeDTDProbMass	121
TRADEMGEN::DemandParserHelper::storeFFCode	123
TRADEMGEN::DemandParserHelper::storeFFProbMass	124
TRADEMGEN::DemandParserHelper::storeOrigin	126
TRADEMGEN::DemandParserHelper::storePosCode	128
TRADEMGEN::DemandParserHelper::storePosProbMass	129
TRADEMGEN::DemandParserHelper::storePrefCabin	131
TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd	132
TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart	134
TRADEMGEN::DemandParserHelper::storePrefDepTime	136
TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass	137
TRADEMGEN::DemandParserHelper::storeStayCode	139
TRADEMGEN::DemandParserHelper::storeStayProbMass	141
TRADEMGEN::DemandParserHelper::storeTimeValue	142
TRADEMGEN::DemandParserHelper::storeTimeValueProbMass	144
TRADEMGEN::DemandParserHelper::storeTripCode	146
TRADEMGEN::DemandParserHelper::storeTripProbMass	147
TRADEMGEN::DemandParserHelper::storeWTP	149
std::priority_queue< T >	
std::queue< T >	
RootException	102
TRADEMGEN::TrademgenGenerationException	165
TRADEMGEN::IndexOutOfRangeException	97
ServiceAbstract	103
TRADEMGEN::TRADEMGEN_ServiceContext	163
std::set< K >	
std::stack< T >	
StructAbstract	151
TRADEMGEN::DemandCharacteristics	45
TRADEMGEN::DemandDistribution	50



<b>TRADEMGEN::DemandStruct</b>	<b>84</b>
<b>TRADEMGEN::RandomGenerationContext</b>	<b>100</b>
<b>TRADEMGEN::TRADEMGEN_Abstract</b>	<b>151</b>
<b>TRADEMGEN::DBParams</b>	<b>42</b>
<b>TRADEMGEN::TRADEMGEN_Service</b>	<b>153</b>
<b>TRADEMGEN::Trademgener</b>	<b>164</b>
std::valarray< T >	
std::vector< T >	

## 5 TraDemGen Class Index

### 5.1 TraDemGen Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<b>BomAbstract</b>	<b>30</b>
<b>TRADEMGEN::BomDisplay</b> (Utility class to display TraDemGen objects with a pretty format )	<b>30</b>
<b>grammar</b>	<b>31</b>
<b>stdair::CategoricalAttribute&lt; T &gt;</b> (Class modeling the distribution of values that can be taken by a categorical attribute )	<b>31</b>
<b>TRADEMGEN::CategoricalAttributeLite&lt; T &gt;</b> (Class modeling the distribution of values that can be taken by a categorical attribute )	<b>33</b>
<b>CmdAbstract</b>	<b>36</b>
<b>TRADEMGEN::ContinuousAttribute&lt; T &gt;</b>	<b>36</b>
<b>TRADEMGEN::ContinuousAttributeLite&lt; T &gt;</b> (Class modeling the distribution of values that can be taken by a continuous attribute )	<b>38</b>
<b>TestFixture</b>	<b>40</b>
<b>TRADEMGEN::DBManager</b>	<b>40</b>
<b>TRADEMGEN::DBParams</b>	<b>42</b>
<b>TRADEMGEN::DefaultMap</b>	<b>45</b>
<b>TRADEMGEN::DemandCharacteristics</b> (Class modeling the characteristics of a demand type )	<b>45</b>
<b>TRADEMGEN::DemandDistribution</b> (Class modeling the distribution of a demand type )	<b>50</b>
<b>TRADEMGEN::DemandFileParser</b>	<b>52</b>

<b>TRADEMGEN::DemandFilePath</b>	<b>53</b>
<b>DemandGenerationTestSuite</b>	<b>54</b>
<b>TRADEMGEN::DemandInputFileNotFoundException</b>	<b>55</b>
<b>TRADEMGEN::DemandManager</b> (Utility class for Demand and DemandStream objects )	<b>55</b>
<b>TRADEMGEN::DemandParser</b> (Class wrapping the parser entry point )	<b>56</b>
<b>TRADEMGEN::DemandParserHelper::DemandParser</b>	<b>57</b>
<b>TRADEMGEN::DemandParserHelper::DemandParser::definition&lt; ScannerT &gt;</b>	<b>59</b>
<b>TRADEMGEN::DemandStream</b> (Class modeling a demand stream )	<b>68</b>
<b>TRADEMGEN::DemandStreamKey</b>	<b>82</b>
<b>TRADEMGEN::DemandStruct</b>	<b>84</b>
<b>TRADEMGEN::DictionaryManager</b> (Class wrapper of dictionary business methods )	<b>91</b>
<b>TRADEMGEN::DemandParserHelper::doEndDemand</b>	<b>92</b>
<b>FacServiceAbstract</b>	<b>94</b>
<b>TRADEMGEN::FacTRADEMGENServiceContext</b> (Factory for creating the TraDemGen service context instance )	<b>94</b>
<b>FileNotFoundException</b>	<b>96</b>
<b>TRADEMGEN::FlagSaver</b>	<b>96</b>
<b>TRADEMGEN::IndexOutOfRangeException</b>	<b>97</b>
<b>InputFilePath</b>	<b>98</b>
<b>KeyAbstract</b>	<b>98</b>
<b>TRADEMGEN::DemandParserHelper::ParserSemanticAction</b>	<b>98</b>
<b>TRADEMGEN::RandomGenerationContext</b>	<b>100</b>
<b>RootException</b>	<b>102</b>
<b>ServiceAbstract</b>	<b>103</b>
<b>TRADEMGEN::DemandParserHelper::storeChannelCode</b>	<b>103</b>
<b>TRADEMGEN::DemandParserHelper::storeChannelProbMass</b>	<b>105</b>
<b>TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility</b>	<b>106</b>
<b>TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb</b>	<b>108</b>
<b>TRADEMGEN::DemandParserHelper::storeDemandMean</b>	<b>110</b>
<b>TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility</b>	<b>111</b>

TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb	113
TRADEMGEN::DemandParserHelper::storeDemandStdDev	115
TRADEMGEN::DemandParserHelper::storeDestination	116
TRADEMGEN::DemandParserHelper::storeDow	118
TRADEMGEN::DemandParserHelper::storeDTD	119
TRADEMGEN::DemandParserHelper::storeDTDProbMass	121
TRADEMGEN::DemandParserHelper::storeFFCode	123
TRADEMGEN::DemandParserHelper::storeFFProbMass	124
TRADEMGEN::DemandParserHelper::storeOrigin	126
TRADEMGEN::DemandParserHelper::storePosCode	128
TRADEMGEN::DemandParserHelper::storePosProbMass	129
TRADEMGEN::DemandParserHelper::storePrefCabin	131
TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd	132
TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart	134
TRADEMGEN::DemandParserHelper::storePrefDepTime	136
TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass	137
TRADEMGEN::DemandParserHelper::storeStayCode	139
TRADEMGEN::DemandParserHelper::storeStayProbMass	141
TRADEMGEN::DemandParserHelper::storeTimeValue	142
TRADEMGEN::DemandParserHelper::storeTimeValueProbMass	144
TRADEMGEN::DemandParserHelper::storeTripCode	146
TRADEMGEN::DemandParserHelper::storeTripProbMass	147
TRADEMGEN::DemandParserHelper::storeWTP	149
StructAbstract	151
TRADEMGEN::TRADEMGEN_Abstract	151
TRADEMGEN::TRADEMGEN_Service (Class holding the services related to Travel Demand Generation )	153
TRADEMGEN::TRADEMGEN_ServiceContext (Class holding the context of the Trademgen services )	163
TRADEMGEN::Trademgener (Wrapper structure around the C++ API, so as to expose a Python API )	164

<a href="#">TRADEMGEN::TrademgenGenerationException</a>	165
---	-----

## 6 TraDemGen File Index

### 6.1 TraDemGen File List

Here is a list of all files with brief descriptions:

<a href="#">test/trademgen/DemandGenerationTestSuite.cpp</a>	167
<a href="#">test/trademgen/DemandGenerationTestSuite.hpp</a>	167
<a href="#">test/trademgen/generateEvents.cpp</a>	168
<a href="#">trademgen/DBParams.hpp</a>	191
<a href="#">trademgen/TRADEMGEN_Abstract.hpp</a>	195
<a href="#">trademgen/TRADEMGEN_Exceptions.hpp</a>	196
<a href="#">trademgen/TRADEMGEN_Service.hpp</a>	197
<a href="#">trademgen/TRADEMGEN_Types.hpp</a>	197
<a href="#">trademgen/basic/BasConst.cpp</a>	168
<a href="#">trademgen/basic/BasConst_DemandGeneration.hpp</a>	169
<a href="#">trademgen/basic/BasConst_TRADEMGEN_Service.hpp</a>	169
<a href="#">trademgen/basic/BasParserTypes.hpp</a>	169
<a href="#">trademgen/basic/CategoricalAttribute.hpp</a>	170
<a href="#">trademgen/basic/CategoricalAttributeLite.hpp</a>	171
<a href="#">trademgen/basic/ContinuousAttribute.hpp</a>	171
<a href="#">trademgen/basic/ContinuousAttributeLite.hpp</a>	171
<a href="#">trademgen/basic/DemandCharacteristics.cpp</a>	172
<a href="#">trademgen/basic/DemandCharacteristics.hpp</a>	172
<a href="#">trademgen/basic/DemandCharacteristicsTypes.hpp</a>	173
<a href="#">trademgen/basic/DemandDistribution.cpp</a>	174
<a href="#">trademgen/basic/DemandDistribution.hpp</a>	174
<a href="#">trademgen/basic/DictionaryManager.cpp</a>	174
<a href="#">trademgen/basic/DictionaryManager.hpp</a>	174
<a href="#">trademgen/basic/RandomGenerationContext.cpp</a>	175

trademgen/basic/ <a href="#">RandomGenerationContext.hpp</a>	175
trademgen/batches/ <a href="#">trademgen_generateDemand.cpp</a>	175
trademgen/batches/ <a href="#">trademgen_with_db.cpp</a>	179
trademgen/bom/ <a href="#">BomDisplay.cpp</a>	183
trademgen/bom/ <a href="#">BomDisplay.hpp</a>	183
trademgen/bom/ <a href="#">DemandStream.cpp</a>	183
trademgen/bom/ <a href="#">DemandStream.hpp</a>	184
trademgen/bom/ <a href="#">DemandStreamKey.cpp</a>	184
trademgen/bom/ <a href="#">DemandStreamKey.hpp</a>	185
trademgen/bom/ <a href="#">DemandStreamTypes.hpp</a>	185
trademgen/bom/ <a href="#">DemandStruct.cpp</a>	185
trademgen/bom/ <a href="#">DemandStruct.hpp</a>	186
trademgen/command/ <a href="#">DBManager.cpp</a>	186
trademgen/command/ <a href="#">DBManager.hpp</a>	186
trademgen/command/ <a href="#">DemandManager.cpp</a>	187
trademgen/command/ <a href="#">DemandManager.hpp</a>	187
trademgen/command/ <a href="#">DemandParser.cpp</a>	188
trademgen/command/ <a href="#">DemandParser.hpp</a>	188
trademgen/command/ <a href="#">DemandParserHelper.cpp</a>	189
trademgen/command/ <a href="#">DemandParserHelper.hpp</a>	190
trademgen/factory/ <a href="#">FacTRADEMGENSEerviceContext.cpp</a>	191
trademgen/factory/ <a href="#">FacTRADEMGENSEerviceContext.hpp</a>	192
trademgen/python/ <a href="#">pytrademgen.cpp</a>	192
trademgen/service/ <a href="#">TRADEMGEN_Service.cpp</a>	193
trademgen/service/ <a href="#">TRADEMGEN_ServiceContext.cpp</a>	195
trademgen/service/ <a href="#">TRADEMGEN_ServiceContext.hpp</a>	195
trademgen/ui/cmdline/ <a href="#">trademgen.cpp</a>	198

## 7 TraDemGen Page Index

### 7.1 TraDemGen Related Pages

Here is a list of all related documentation pages:

<b>People</b>	<a href="#">198</a>
<b>Coding Rules</b>	<a href="#">198</a>
<b>Copyright and License</b>	<a href="#">199</a>
<b>Documentation Rules</b>	<a href="#">206</a>
<b>Main features</b>	<a href="#">208</a>
<b>Make a Difference</b>	<a href="#">208</a>
<b>Make a new release</b>	<a href="#">209</a>
<b>Installation</b>	<a href="#">212</a>
<b>Linking with TraDemGen</b>	<a href="#">222</a>
<b>Test Rules</b>	<a href="#">223</a>
<b>Users Guide</b>	<a href="#">224</a>
<b>Supported Systems</b>	<a href="#">225</a>
<b>TraDemGen Supported Systems (Previous Releases)</b>	<a href="#">233</a>
<b>Tutorials</b>	<a href="#">233</a>
<b>Command-Line Test to Demonstrate How To Use TraDemGen elements</b>	<a href="#">236</a>

## 8 TraDemGen Directory Documentation

### 8.1 trademgen/basic/ Directory Reference

#### Files

- file [BasConst.cpp](#)
- file [BasConst\\_DemandGeneration.hpp](#)
- file [BasConst\\_TRADEMGEN\\_Service.hpp](#)
- file [BasParserTypes.hpp](#)
- file [CategoricalAttribute.hpp](#)
- file [CategoricalAttributeLite.hpp](#)
- file [ContinuousAttribute.hpp](#)
- file [ContinuousAttributeLite.hpp](#)
- file [DemandCharacteristics.cpp](#)
- file [DemandCharacteristics.hpp](#)
- file [DemandCharacteristicsTypes.hpp](#)

- file [DemandDistribution.cpp](#)
- file [DemandDistribution.hpp](#)
- file [DictionaryManager.cpp](#)
- file [DictionaryManager.hpp](#)
- file [RandomGenerationContext.cpp](#)
- file [RandomGenerationContext.hpp](#)

## 8.2 trademgen/batches/ Directory Reference

### Files

- file [trademgen\\_generateDemand.cpp](#)
- file [trademgen\\_with\\_db.cpp](#)

## 8.3 trademgen/bom/ Directory Reference

### Files

- file [BomDisplay.cpp](#)
- file [BomDisplay.hpp](#)
- file [DemandStream.cpp](#)
- file [DemandStream.hpp](#)
- file [DemandStreamKey.cpp](#)
- file [DemandStreamKey.hpp](#)
- file [DemandStreamTypes.hpp](#)
- file [DemandStruct.cpp](#)
- file [DemandStruct.hpp](#)

## 8.4 trademgen/ui/cmdline/ Directory Reference

### Files

- file [trademgen.cpp](#)

## 8.5 trademgen/command/ Directory Reference

### Files

- file [DBManager.cpp](#)
- file [DBManager.hpp](#)
- file [DemandManager.cpp](#)
- file [DemandManager.hpp](#)
- file [DemandParser.cpp](#)
- file [DemandParser.hpp](#)
- file [DemandParserHelper.cpp](#)
- file [DemandParserHelper.hpp](#)

## 8.6 trademgen/factory/ Directory Reference

### Files

- file [FacTRADEMGENServiceContext.cpp](#)
- file [FacTRADEMGENServiceContext.hpp](#)

## 8.7 trademgen/python/ Directory Reference

### Files

- file [pytrademgen.cpp](#)

## 8.8 trademgen/service/ Directory Reference

### Files

- file [TRADEMGEN\\_Service.cpp](#)
- file [TRADEMGEN\\_ServiceContext.cpp](#)
- file [TRADEMGEN\\_ServiceContext.hpp](#)

## 8.9 test/ Directory Reference

### Directories

- directory [trademgen](#)

## 8.10 trademgen/ Directory Reference

### Directories

- directory [basic](#)
- directory [batches](#)
- directory [bom](#)
- directory [command](#)
- directory [factory](#)
- directory [python](#)
- directory [service](#)
- directory [ui](#)

### Files

- file [DBParams.hpp](#)
- file [TRADEMGEN\\_Abstract.hpp](#)
- file [TRADEMGEN\\_Exceptions.hpp](#)
- file [TRADEMGEN\\_Service.hpp](#)
- file [TRADEMGEN\\_Types.hpp](#)



## 8.11 test/trademgen/ Directory Reference

### Files

- file [DemandGenerationTestSuite.cpp](#)
- file [DemandGenerationTestSuite.hpp](#)
- file [generateEvents.cpp](#)

## 8.12 trademgen/ui/ Directory Reference

### Directories

- directory [cmdline](#)

# 9 TraDemGen Namespace Documentation

## 9.1 SEVMGR Namespace Reference

### Functions

- template void [SEVMGR\\_Service::addEventGenerator< TRADEMGEN::DemandStream >](#) ([TRADEMGEN::DemandStream](#) &) const
- template [TRADEMGEN::DemandStream](#) & [SEVMGR\\_Service::getEventGenerator< TRADEMGEN::DemandStream, stdair::DemandStreamKeyStr\\_T >](#) (const stdair::DemandStreamKeyStr\_T &) const
- template bool [SEVMGR\\_Service::hasEventGenerator< TRADEMGEN::DemandStream, stdair::DemandStreamKeyStr\\_T >](#) (const stdair::DemandStreamKeyStr\_T &) const
- template const [TRADEMGEN::DemandStreamList\\_T](#) [SEVMGR\\_Service::getEventGeneratorList< TRADEMGEN::DemandStream >](#) () const
- template bool [SEVMGR\\_Service::hasEventGeneratorList< TRADEMGEN::DemandStream >](#) () const

### 9.1.1 Function Documentation

**9.1.1.1** template void [SEVMGR::SEVMGR\\_Service::addEventGenerator< TRADEMGEN::DemandStream >](#) ([TRADEMGEN::DemandStream](#) &) const

Explicit template instantiations with the TraDemGen own EventGenerator type: DemandStream.

**9.1.1.2** template [TRADEMGEN::DemandStream&](#) [SEVMGR::SEVMGR\\_Service::getEventGenerator< TRADEMGEN::DemandStream, stdair::DemandStreamKeyStr\\_T >](#) (const stdair::DemandStreamKeyStr\_T &) const

**9.1.1.3** template bool [SEVMGR::SEVMGR\\_Service::hasEventGenerator< TRADEMGEN::DemandStream, stdair::DemandStreamKeyStr\\_T >](#) (const stdair::DemandStreamKeyStr\_T &) const

**9.1.1.4** template const [TRADEMGEN::DemandStreamList\\_T](#) [SEVMGR::SEVMGR\\_Service::getEventGeneratorList< TRADEMGEN::DemandStream >](#) () const

9.1.1.5 template bool SEVMGR::SEVMGR\_Service::hasEventGeneratorList<  
[TRADEMGEN::DemandStream](#) > () const

## 9.2 stdair Namespace Reference

Forward declarations.

### Classes

- struct [CategoricalAttribute](#)  
*Class modeling the distribution of values that can be taken by a categorical attribute.*

### 9.2.1 Detailed Description

Forward declarations.

## 9.3 TRADEMGEN Namespace Reference

### Classes

- struct [DefaultMap](#)
- struct [CategoricalAttributeLite](#)  
*Class modeling the distribution of values that can be taken by a categorical attribute.*
- struct [ContinuousAttribute](#)
- struct [ContinuousAttributeLite](#)  
*Class modeling the distribution of values that can be taken by a continuous attribute.*
- struct [DemandCharacteristics](#)  
*Class modeling the characteristics of a demand type.*
- struct [DemandDistribution](#)  
*Class modeling the distribution of a demand type.*
- class [DictionaryManager](#)  
*Class wrapper of dictionary business methods.*
- struct [RandomGenerationContext](#)
- struct [FlagSaver](#)
- class [BomDisplay](#)  
*Utility class to display TraDemGen objects with a pretty format.*
- class [DemandStream](#)  
*Class modeling a demand stream.*
- struct [DemandStreamKey](#)
- struct [DemandStruct](#)

- class [DBManager](#)
- class [DemandManager](#)  
*Utility class for Demand and [DemandStream](#) objects.*
- class [DemandParser](#)  
*Class wrapping the parser entry point.*
- class [DemandFileParser](#)
- struct [DBParams](#)
- class [FacTRADEMGENSEerviceContext](#)  
*Factory for creating the TraDemGen service context instance.*
- struct [Trademgener](#)  
*Wrapper structure around the C++ API, so as to expose a Python API.*
- class [TRADEMGEN\\_ServiceContext](#)  
*Class holding the context of the Trademgen services.*
- struct [TRADEMGEN\\_Abstract](#)
- class [TrademgenGenerationException](#)
- class [DemandInputFileNotFoundException](#)
- class [IndexOutOfRangeException](#)
- class [TRADEMGEN\\_Service](#)  
*class holding the services related to Travel Demand Generation.*
- class [DemandFilePath](#)

## Namespaces

- namespace [DemandParserHelper](#)

## Typedefs

- typedef char [char\\_t](#)
- typedef boost::spirit::classic::file\_iterator< [char\\_t](#) > [iterator\\_t](#)
- typedef boost::spirit::classic::scanner< [iterator\\_t](#) > [scanner\\_t](#)
- typedef boost::spirit::classic::rule< [scanner\\_t](#) > [rule\\_t](#)
- typedef boost::spirit::classic::int\_parser< unsigned int, 10, 1, 1 > [int1\\_p\\_t](#)
- typedef boost::spirit::classic::uint\_parser< unsigned int, 10, 2, 2 > [uint2\\_p\\_t](#)
- typedef boost::spirit::classic::uint\_parser< unsigned int, 10, 1, 2 > [uint1\\_2\\_p\\_t](#)
- typedef boost::spirit::classic::uint\_parser< unsigned int, 10, 1, 3 > [uint1\\_3\\_p\\_t](#)
- typedef boost::spirit::classic::uint\_parser< unsigned int, 10, 4, 4 > [uint4\\_p\\_t](#)
- typedef boost::spirit::classic::uint\_parser< unsigned int, 10, 1, 4 > [uint1\\_4\\_p\\_t](#)
- typedef boost::spirit::classic::chset< [char\\_t](#) > [chset\\_t](#)
- typedef boost::spirit::classic::impl::loop\_traits< [chset\\_t](#), unsigned int, unsigned int >::type [repeat\\_p\\_t](#)
- typedef boost::spirit::classic::bounded< [uint2\\_p\\_t](#), unsigned int > [bounded2\\_p\\_t](#)
- typedef boost::spirit::classic::bounded< [uint1\\_2\\_p\\_t](#), unsigned int > [bounded1\\_2\\_p\\_t](#)
- typedef boost::spirit::classic::bounded< [uint1\\_3\\_p\\_t](#), unsigned int > [bounded1\\_3\\_p\\_t](#)
- typedef boost::spirit::classic::bounded< [uint4\\_p\\_t](#), unsigned int > [bounded4\\_p\\_t](#)

- typedef boost::spirit::classic::bounded< uint1\_4\_p\_t, unsigned int > bounded1\_4\_p\_t
- typedef ContinuousAttributeLite< stdair::FloatDuration\_T > ContinuousFloatDuration\_T
- typedef ContinuousFloatDuration\_T::ContinuousDistribution\_T ArrivalPatternCumulativeDistribution\_T
- typedef CategoricalAttributeLite< stdair::AirportCode\_T > POSProbabilityMass\_T
- typedef POSProbabilityMass\_T::ProbabilityMassFunction\_T POSProbabilityMassFunction\_T
- typedef CategoricalAttributeLite< stdair::ChannelLabel\_T > ChannelProbabilityMass\_T
- typedef ChannelProbabilityMass\_T::ProbabilityMassFunction\_T ChannelProbabilityMassFunction\_T
- typedef CategoricalAttributeLite< stdair::TripType\_T > TripTypeProbabilityMass\_T
- typedef TripTypeProbabilityMass\_T::ProbabilityMassFunction\_T TripTypeProbabilityMassFunction\_T
- typedef CategoricalAttributeLite< stdair::DayDuration\_T > StayDurationProbabilityMass\_T
- typedef StayDurationProbabilityMass\_T::ProbabilityMassFunction\_T StayDurationProbabilityMassFunction\_T
- typedef CategoricalAttributeLite< stdair::FrequentFlyer\_T > FrequentFlyerProbabilityMass\_T
- typedef FrequentFlyerProbabilityMass\_T::ProbabilityMassFunction\_T FrequentFlyerProbabilityMassFunction\_T
- typedef ContinuousAttributeLite< stdair::IntDuration\_T > PreferredDepartureTimeCumulativeDistribution\_T
- typedef PreferredDepartureTimeCumulativeDistribution\_T::ContinuousDistribution\_T PreferredDepartureTimeContinuousDistribution\_T
- typedef ContinuousAttributeLite< stdair::PriceValue\_T > ValueOfTimeCumulativeDistribution\_T
- typedef ValueOfTimeCumulativeDistribution\_T::ContinuousDistribution\_T ValueOfTimeContinuousDistribution\_T
- typedef ContinuousAttributeLite< stdair::RealNumber\_T > CumulativeDistribution\_T
- typedef CumulativeDistribution\_T::ContinuousDistribution\_T FRAT5Pattern\_T
- typedef stdair::Probability\_T DictionaryKey\_T
- typedef std::list< DemandStream \* > DemandStreamList\_T
- typedef std::map< const stdair::MapKey\_T, DemandStream \* > DemandStreamMap\_T
- typedef std::list< std::string > DBParamsNameList\_T
- typedef boost::shared\_ptr< TRADEMGEN\_Service > TRADEMGEN\_ServicePtr\_T

## Functions

- stdair::BaseGenerator\_T DEFAULT\_BASE\_GENERATOR (stdair::DEFAULT\_RANDOM\_SEED)
- stdair::UniformGenerator\_T DEFAULT\_UNIFORM\_GENERATOR (DEFAULT\_BASE\_GENERATOR, DEFAULT\_UNIFORM\_REAL\_DISTRIBUTION)
- void stat\_display (std::ostream &oStream, const stat\_acc\_type &iStatAcc)

## Variables

- const POSProbabilityMassFunction\_T DEFAULT\_POS\_PROBALILITY\_MASS
- const stdair::FloatDuration\_T DEFAULT\_LAST\_LOWER\_BOUND\_ARRIVAL\_PATTERN = -1
- const FRAT5Pattern\_T DEFAULT\_FRAT5\_PATTERN = DefaultMap::createFRAT5Pattern()
- const double DEFAULT\_MAX\_ADVANCE\_PURCHASE = 330.0
- const stdair::UniformDistribution\_T DEFAULT\_UNIFORM\_REAL\_DISTRIBUTION
- const POSProbabilityMassFunction\_T DEFAULT\_POS\_PROBALILITY\_MASS
- const FRAT5Pattern\_T DEFAULT\_FRAT5\_PATTERN
- const stdair::FloatDuration\_T DEFAULT\_LAST\_LOWER\_BOUND\_ARRIVAL\_PATTERN

- const double [DEFAULT\\_MAX\\_ADVANCE\\_PURCHASE](#)
- stdair::BaseGenerator\_T [DEFAULT\\_BASE\\_GENERATOR](#)
- stdair::UniformGenerator\_T [DEFAULT\\_UNIFORM\\_GENERATOR](#)
- const stdair::UniformDistribution\_T [DEFAULT\\_UNIFORM\\_REAL\\_DISTRIBUTION](#)

### 9.3.1 Typedef Documentation

#### 9.3.1.1 typedef char [TRADEMGENT::char\\_t](#)

Definition at line 31 of file BasParserTypes.hpp.

#### 9.3.1.2 typedef boost::spirit::classic::file\_iterator<[char\\_t](#)> [TRADEMGENT::iterator\\_t](#)

Definition at line 35 of file BasParserTypes.hpp.

#### 9.3.1.3 typedef boost::spirit::classic::scanner<[iterator\\_t](#)> [TRADEMGENT::scanner\\_t](#)

Definition at line 36 of file BasParserTypes.hpp.

#### 9.3.1.4 typedef boost::spirit::classic::rule<[scanner\\_t](#)> [TRADEMGENT::rule\\_t](#)

Definition at line 37 of file BasParserTypes.hpp.

#### 9.3.1.5 typedef boost::spirit::classic::int\_parser<unsigned int, 10, 1, 1> [TRADEMGENT::int1\\_p\\_t](#)

1-digit-integer parser

Definition at line 45 of file BasParserTypes.hpp.

#### 9.3.1.6 typedef boost::spirit::classic::uint\_parser<unsigned int, 10, 2, 2> [TRADEMGENT::uint2\\_p\\_t](#)

2-digit-integer parser

Definition at line 48 of file BasParserTypes.hpp.

#### 9.3.1.7 typedef boost::spirit::classic::uint\_parser<unsigned int, 10, 1, 2> [TRADEMGENT::uint1\\_2\\_p\\_t](#)

Up-to-2-digit-integer parser

Definition at line 51 of file BasParserTypes.hpp.

#### 9.3.1.8 typedef boost::spirit::classic::uint\_parser<unsigned int, 10, 1, 3> [TRADEMGENT::uint1\\_3\\_p\\_t](#)

Up-to-3-digit-integer parser

Definition at line 54 of file BasParserTypes.hpp.

**9.3.1.9** `typedef boost::spirit::classic::uint_parser<unsigned int, 10, 4, 4> TRADEMGEN::uint4_p_t`

4-digit-integer parser

Definition at line 57 of file BasParserTypes.hpp.

**9.3.1.10** `typedef boost::spirit::classic::uint_parser<unsigned int, 10, 1, 4> TRADEMGEN::uint1_4_p_t`

Up-to-4-digit-integer parser

Definition at line 60 of file BasParserTypes.hpp.

**9.3.1.11** `typedef boost::spirit::classic::chset<char_t> TRADEMGEN::chset_t`

character set

Definition at line 63 of file BasParserTypes.hpp.

**9.3.1.12** `typedef boost::spirit::classic::impl::loop_traits<chset_t, unsigned int, unsigned int>::type TRADEMGEN::repeat_p_t`

(Repeating) sequence of a given number of characters: repeat\_p(min, max)

Definition at line 69 of file BasParserTypes.hpp.

**9.3.1.13** `typedef boost::spirit::classic::bounded<uint2_p_t, unsigned int> TRADEMGEN::bounded2_p_t`

Bounded-number-of-integers parser

Definition at line 72 of file BasParserTypes.hpp.

**9.3.1.14** `typedef boost::spirit::classic::bounded<uint1_2_p_t, unsigned int> TRADEMGEN::bounded1_2_p_t`

Definition at line 73 of file BasParserTypes.hpp.

**9.3.1.15** `typedef boost::spirit::classic::bounded<uint1_3_p_t, unsigned int> TRADEMGEN::bounded1_3_p_t`

Definition at line 74 of file BasParserTypes.hpp.

**9.3.1.16** `typedef boost::spirit::classic::bounded<uint4_p_t, unsigned int> TRADEMGEN::bounded4_p_t`

Definition at line 75 of file BasParserTypes.hpp.

**9.3.1.17** `typedef boost::spirit::classic::bounded<uint1_4_p_t, unsigned int> TRADEMGEN::bounded1_4_p_t`

Definition at line 76 of file BasParserTypes.hpp.

**9.3.1.18** `typedef` [ContinuousAttributeLite<stdair::FloatDuration\\_T>](#)  
[TRADEMGENT::ContinuousFloatDuration\\_T](#)

Type definition for the continuous distribution of the duration (as a float number).

Definition at line 19 of file DemandCharacteristicsTypes.hpp.

**9.3.1.19** `typedef` [ContinuousFloatDuration\\_T::ContinuousDistribution\\_T](#)  
[TRADEMGENT::ArrivalPatternCumulativeDistribution\\_T](#)

Type definition for the arrival pattern cumulative distribution.

Definition at line 22 of file DemandCharacteristicsTypes.hpp.

**9.3.1.20** `typedef` [CategoricalAttributeLite<stdair::AirportCode\\_T>](#)  
[TRADEMGENT::POSProbabilityMass\\_T](#)

Define the point-of-sale probability mass.

Definition at line 25 of file DemandCharacteristicsTypes.hpp.

**9.3.1.21** `typedef` [POSProbabilityMass\\_T::ProbabilityMassFunction\\_T](#)  
[TRADEMGENT::POSProbabilityMassFunction\\_T](#)

Define the probability mass function type of point-of-sale.

Definition at line 28 of file DemandCharacteristicsTypes.hpp.

**9.3.1.22** `typedef` [CategoricalAttributeLite<stdair::ChannelLabel\\_T>](#) [TRADEMGENT::Channel-  
ProbabilityMass\\_T](#)

Define the booking channel probability mass.

Definition at line 31 of file DemandCharacteristicsTypes.hpp.

**9.3.1.23** `typedef` [ChannelProbabilityMass\\_T::ProbabilityMassFunction\\_T](#)  
[TRADEMGENT::ChannelProbabilityMassFunction\\_T](#)

Define the probability mass function type of booking channel.

Definition at line 34 of file DemandCharacteristicsTypes.hpp.

**9.3.1.24** `typedef` [CategoricalAttributeLite<stdair::TripType\\_T>](#) [TRADEMGENT::TripType-  
ProbabilityMass\\_T](#)

Define the trip type probability mass.

Definition at line 37 of file DemandCharacteristicsTypes.hpp.

**9.3.1.25** `typedef` [TripTypeProbabilityMass\\_T::ProbabilityMassFunction\\_T](#) [TRADEMGENT::Trip-  
TypeProbabilityMassFunction\\_T](#)

Define the probability mass function type of trip type.

Definition at line 40 of file DemandCharacteristicsTypes.hpp.

**9.3.1.26** `typedef` [CategoricalAttributeLite<stdair::DayDuration\\_T>](#) [TRADEMGENT::StayDurationProbabilityMass\\_T](#)

Define the stay duration probability mass.

Definition at line 43 of file DemandCharacteristicsTypes.hpp.

**9.3.1.27** `typedef` [StayDurationProbabilityMass\\_T::ProbabilityMassFunction\\_T](#) [TRADEMGENT::StayDurationProbabilityMassFunction\\_T](#)

Define the probability mass function type of stay duration.

Definition at line 46 of file DemandCharacteristicsTypes.hpp.

**9.3.1.28** `typedef` [CategoricalAttributeLite<stdair::FrequentFlyer\\_T>](#) [TRADEMGENT::FrequentFlyerProbabilityMass\\_T](#)

Define the frequent flyer probability mass.

Definition at line 49 of file DemandCharacteristicsTypes.hpp.

**9.3.1.29** `typedef` [FrequentFlyerProbabilityMass\\_T::ProbabilityMassFunction\\_T](#) [TRADEMGENT::FrequentFlyerProbabilityMassFunction\\_T](#)

Define the probability mass function type of frequent flyer.

Definition at line 52 of file DemandCharacteristicsTypes.hpp.

**9.3.1.30** `typedef` [ContinuousAttributeLite<stdair::IntDuration\\_T>](#) [TRADEMGENT::PreferredDepartureTimeCumulativeDistribution\\_T](#)

Define the preferred departure time cumulative distribution.

Definition at line 55 of file DemandCharacteristicsTypes.hpp.

**9.3.1.31** `typedef` [PreferredDepartureTimeCumulativeDistribution\\_T::ContinuousDistribution\\_T](#) [TRADEMGENT::PreferredDepartureTimeContinuousDistribution\\_T](#)

Define the preferred departure time continuous distribution.

Definition at line 58 of file DemandCharacteristicsTypes.hpp.

**9.3.1.32** `typedef` [ContinuousAttributeLite<stdair::PriceValue\\_T>](#) [TRADEMGENT::ValueOfTimeCumulativeDistribution\\_T](#)

Define the value of time cumulative distribution.

Definition at line 61 of file DemandCharacteristicsTypes.hpp.

**9.3.1.33** `typedef` [ValueOfTimeCumulativeDistribution\\_T::ContinuousDistribution\\_T](#) [TRADEMGENT::ValueOfTimeContinuousDistribution\\_T](#)

Define the value of time continuous distribution.

Definition at line 64 of file DemandCharacteristicsTypes.hpp.



**9.3.1.34** typedef [ContinuousAttributeLite](#)<stdair::RealNumber\_T> [TRADEMGEN::CumulativeDistribution\\_T](#)

Define the FRAT5 pattern type.

Definition at line 67 of file DemandCharacteristicsTypes.hpp.

**9.3.1.35** typedef [CumulativeDistribution\\_T::ContinuousDistribution\\_T](#) [TRADEMGEN::FRAT5Pattern\\_T](#)

Definition at line 68 of file DemandCharacteristicsTypes.hpp.

**9.3.1.36** typedef stdair::Probability\_T [TRADEMGEN::DictionaryKey\\_T](#)

Dictionary key.

Definition at line 16 of file DictionaryManager.hpp.

**9.3.1.37** typedef std::list<[DemandStream\\*](#)> [TRADEMGEN::DemandStreamList\\_T](#)

Define the airline feature list.

Definition at line 16 of file DemandStreamTypes.hpp.

**9.3.1.38** typedef std::map<const stdair::MapKey\_T, [DemandStream\\*](#)> [TRADEMGEN::DemandStreamMap\\_T](#)

Define the airline feature map.

Definition at line 22 of file DemandStreamTypes.hpp.

**9.3.1.39** typedef std::list<std::string> [TRADEMGEN::DBParamsNameList\\_T](#)

List of names for a given (geographical) dbparams.

Definition at line 17 of file DBParams.hpp.

**9.3.1.40** typedef boost::shared\_ptr<[TRADEMGEN\\_Service](#)> [TRADEMGEN::TRADEMGEN\\_ServicePtr\\_T](#)

(Smart) Pointer on the TraDemGen service handler.

Definition at line 17 of file TRADEMGEN\_Types.hpp.

**9.3.2 Function Documentation****9.3.2.1** stdair::BaseGenerator\_T [TRADEMGEN::DEFAULT\\_BASE\\_GENERATOR](#)  
(stdair::DEFAULT\_RANDOM\_SEED)

Default base generator.

**9.3.2.2** stdair::UniformGenerator\_T [TRADEMGEN::DEFAULT\\_UNIFORM\\_GENERATOR](#)  
([DEFAULT\\_BASE\\_GENERATOR](#), [DEFAULT\\_UNIFORM\\_REAL\\_DISTRIBUTION](#))

Default uniform variate generator.

**9.3.2.3 void TRADEMGEN::stat\_display (std::ostream & *oStream*, const stat\_acc\_type & *iStatAcc*)**

Display the statistics held by the dedicated accumulator.

Definition at line 47 of file pytrademgen.cpp.

Referenced by TRADEMGEN::Trademgener::trademgen().

**9.3.3 Variable Documentation****9.3.3.1 const POSProbabilityMassFunction\_T TRADEMGEN::DEFAULT\_POS\_PROBALILITY\_MASS**

**Initial value:**

```
DefaultMap::createPOSProbMass()
```

Default PoS probability mass.

Definition at line 16 of file BasConst.cpp.

**9.3.3.2 const stdair::FloatDuration\_T TRADEMGEN::DEFAULT\_LAST\_LOWER\_BOUND\_ARRIVAL\_PATTERN = -1**

Default last lower bound of daily rate interval in arrival pattern.

Definition at line 35 of file BasConst.cpp.

Referenced by TRADEMGEN::DemandStream::generateTimeOfRequestPoissonProcess().

**9.3.3.3 const FRAT5Pattern\_T TRADEMGEN::DEFAULT\_FRAT5\_PATTERN = DefaultMap::createFRAT5Pattern()**

Default FRAT5 pattern.

Definition at line 38 of file BasConst.cpp.

**9.3.3.4 const double TRADEMGEN::DEFAULT\_MAX\_ADVANCE\_PURCHASE = 330.0**

Default MAX Advance Purchase.

Definition at line 75 of file BasConst.cpp.

**9.3.3.5 const stdair::UniformDistribution\_T TRADEMGEN::DEFAULT\_UNIFORM\_REAL\_DISTRIBUTION**

Default random uniform real distribution.

Definition at line 81 of file BasConst.cpp.

**9.3.3.6 const POSProbabilityMassFunction\_T TRADEMGEN::DEFAULT\_POS\_PROBALILITY\_MASS**

Default PoS probability mass.

Definition at line 16 of file BasConst.cpp.

**9.3.3.7 const [FRAT5Pattern\\_T](#) TRADEMGEN::DEFAULT\_FRAT5\_PATTERN**

Default FRAT5 pattern.

Definition at line 38 of file BasConst.cpp.

**9.3.3.8 const [stdair::FloatDuration\\_T](#) TRADEMGEN::DEFAULT\_LAST\_LOWER\_BOUND\_ARRIVAL\_PATTERN**

Default last lower bound of daily rate interval in arrival pattern.

Definition at line 35 of file BasConst.cpp.

Referenced by TRADEMGEN::DemandStream::generateTimeOfRequestPoissonProcess().

**9.3.3.9 const double [TRADEMGENT::DEFAULT\\_MAX\\_ADVANCE\\_PURCHASE](#)**

Default MAX Advance Purchase.

Definition at line 75 of file BasConst.cpp.

**9.3.3.10 [stdair::BaseGenerator\\_T](#) TRADEMGENT::DEFAULT\_BASE\_GENERATOR**

Default base generator. Just here to initialise objects (e.g., [stdair::RandomGeneration](#)) with default generator. They are then replaced by a generator, for which the state can better be tracked/stored.

**9.3.3.11 [stdair::UniformGenerator\\_T](#) TRADEMGENT::DEFAULT\_UNIFORM\_GENERATOR**

Default uniform generator. Just here to initialise objects (e.g., [stdair::RandomGeneration](#)) with default generator. They are then replaced by a generator, for which the state can better be tracked/stored.

**9.3.3.12 const [stdair::UniformDistribution\\_T](#) TRADEMGENT::DEFAULT\_UNIFORM\_REAL\_DISTRIBUTION**

Default random uniform real distribution.

Definition at line 81 of file BasConst.cpp.

**9.4 TRADEMGEN::DemandParserHelper Namespace Reference****Classes**

- struct [ParserSemanticAction](#)
- struct [storePrefDepDateRangeStart](#)
- struct [storePrefDepDateRangeEnd](#)
- struct [storeDow](#)
- struct [storeOrigin](#)
- struct [storeDestination](#)
- struct [storePrefCabin](#)
- struct [storeDemandMean](#)
- struct [storeDemandStdDev](#)
- struct [storeDemandChangeFeeProb](#)
- struct [storeDemandChangeFeeDisutility](#)
- struct [storeDemandNonRefundableProb](#)

- struct [storeDemandNonRefundableDisutility](#)
- struct [storePosCode](#)
- struct [storePosProbMass](#)
- struct [storeChannelCode](#)
- struct [storeChannelProbMass](#)
- struct [storeTripCode](#)
- struct [storeTripProbMass](#)
- struct [storeStayCode](#)
- struct [storeStayProbMass](#)
- struct [storeFFCode](#)
- struct [storeFFProbMass](#)
- struct [storePrefDepTime](#)
- struct [storePrefDepTimeProbMass](#)
- struct [storeWTP](#)
- struct [storeTimeValue](#)
- struct [storeTimeValueProbMass](#)
- struct [storeDTD](#)
- struct [storeDTDProbMass](#)
- struct [doEndDemand](#)
- struct [DemandParser](#)

### Functions

- [repeat\\_p\\_t airline\\_code\\_p](#) ([chset\\_t](#)("0-9A-Z").derived(), 2, 3)
- [bounded1\\_4\\_p\\_t flight\\_number\\_p](#) ([uint1\\_4\\_p](#).derived(), 0u, 9999u)
- [bounded4\\_p\\_t year\\_p](#) ([uint4\\_p](#).derived(), 2000u, 2099u)
- [bounded2\\_p\\_t month\\_p](#) ([uint2\\_p](#).derived(), 1u, 12u)
- [bounded2\\_p\\_t day\\_p](#) ([uint2\\_p](#).derived(), 1u, 31u)
- [repeat\\_p\\_t dow\\_p](#) ([chset\\_t](#)("0-1").derived().derived(), 7, 7)
- [repeat\\_p\\_t airport\\_p](#) ([chset\\_t](#)("0-9A-Z").derived(), 3, 3)
- [bounded1\\_2\\_p\\_t hours\\_p](#) ([uint1\\_2\\_p](#).derived(), 0u, 23u)
- [bounded2\\_p\\_t minutes\\_p](#) ([uint2\\_p](#).derived(), 0u, 59u)
- [bounded2\\_p\\_t seconds\\_p](#) ([uint2\\_p](#).derived(), 0u, 59u)
- [chset\\_t cabin\\_code\\_p](#) ("A-Z")
- [chset\\_t passenger\\_type\\_p](#) ("A-Z")
- [chset\\_t ff\\_type\\_p](#) ("A-Z")
- [repeat\\_p\\_t class\\_code\\_list\\_p](#) ([chset\\_t](#)("A-Z").derived(), 1, 26)
- [bounded1\\_3\\_p\\_t stay\\_duration\\_p](#) ([uint1\\_3\\_p](#).derived(), 0u, 999u)

### Variables

- [int1\\_p\\_t int1\\_p](#)
- [uint2\\_p\\_t uint2\\_p](#)
- [uint1\\_2\\_p\\_t uint1\\_2\\_p](#)
- [uint1\\_3\\_p\\_t uint1\\_3\\_p](#)
- [uint4\\_p\\_t uint4\\_p](#)
- [uint1\\_4\\_p\\_t uint1\\_4\\_p](#)
- [int1\\_p\\_t family\\_code\\_p](#)

### 9.4.1 Function Documentation

**9.4.1.1** [repeat\\_p\\_t](#) TRADEMGEN::DemandParserHelper::airline\_code\_p ([chset\\_t](#)("0-9A-Z").derived(), 2, 3)

Airline Code Parser: repeat\_p(2,3)[chset\_p("0-9A-Z")]

**9.4.1.2** [bounded1\\_4\\_p\\_t](#) TRADEMGEN::DemandParserHelper::flight\_number\_p (uint1\_4\_p. derived(), 0u, 9999u)

Flight Number Parser: limit\_d(0u, 9999u)[uint1\_4\_p]

**9.4.1.3** [bounded4\\_p\\_t](#) TRADEMGEN::DemandParserHelper::year\_p (uint4\_p. derived(), 2000u, 2099u)

Year Parser: limit\_d(2000u, 2099u)[uint4\_p]

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**9.4.1.4** [bounded2\\_p\\_t](#) TRADEMGEN::DemandParserHelper::month\_p (uint2\_p. derived(), 1u, 12u)

Month Parser: limit\_d(1u, 12u)[uint2\_p]

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**9.4.1.5** [bounded2\\_p\\_t](#) TRADEMGEN::DemandParserHelper::day\_p (uint2\_p. derived(), 1u, 31u)

Day Parser: limit\_d(1u, 31u)[uint2\_p]

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**9.4.1.6** [repeat\\_p\\_t](#) TRADEMGEN::DemandParserHelper::dow\_p ([chset\\_t](#)("0-1").derived().derived(), 7, 7)

DOW (Day-Of-the-Week) Parser: repeat\_p(7)[chset\_p("0-1")]

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**9.4.1.7** [repeat\\_p\\_t](#) TRADEMGEN::DemandParserHelper::airport\_p ([chset\\_t](#)("0-9A-Z").derived(), 3, 3)

Airport Parser: repeat\_p(3)[chset\_p("0-9A-Z")]

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**9.4.1.8** [bounded1\\_2\\_p\\_t](#) TRADEMGEN::DemandParserHelper::hours\_p (uint1\_2\_p. derived(), 0u, 23u)

Hour Parser: limit\_d(0u, 23u)[uint2\_p]

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

#### 9.4.1.9 [bounded2\\_p\\_t](#) TRADEMGEN::DemandParserHelper::minutes\_p (uint2\_p. *derived()*, 0u, 59u)

Minute Parser: limit\_d(0u, 59u)[uint2\_p]

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

#### 9.4.1.10 [bounded2\\_p\\_t](#) TRADEMGEN::DemandParserHelper::seconds\_p (uint2\_p. *derived()*, 0u, 59u)

Second Parser: limit\_d(0u, 59u)[uint2\_p]

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

#### 9.4.1.11 [chset\\_t](#) TRADEMGEN::DemandParserHelper::cabin\_code\_p ("A-Z")

Cabin code parser: chset\_p("A-Z")

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

#### 9.4.1.12 [chset\\_t](#) TRADEMGEN::DemandParserHelper::passenger\_type\_p ("A-Z")

Passenger type parser: chset\_p("A-Z")

#### 9.4.1.13 [chset\\_t](#) TRADEMGEN::DemandParserHelper::ff\_type\_p ("A-Z")

Frequent flyer type parser: chset\_p("A-Z")

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

#### 9.4.1.14 [repeat\\_p\\_t](#) TRADEMGEN::DemandParserHelper::class\_code\_list\_p ([chset\\_t](#)("A-Z").*derived()*, 1, 26)

Class Code List Parser: repeat\_p(1,26)[chset\_p("A-Z")]

#### 9.4.1.15 [bounded1\\_3\\_p\\_t](#) TRADEMGEN::DemandParserHelper::stay\_duration\_p (uint1\_3\_p. *derived()*, 0u, 999u)

Stay duration Parser: limit\_d(0u, 999u)[uint3\_p]

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

### 9.4.2 Variable Documentation

#### 9.4.2.1 [int1\\_p\\_t](#) TRADEMGEN::DemandParserHelper::int1\_p

1-digit-integer parser

Definition at line 497 of file DemandParserHelper.cpp.

#### 9.4.2.2 [uint2\\_p\\_t TRADEMGEN::DemandParserHelper::uint2\\_p](#)

2-digit-integer parser

Definition at line 500 of file DemandParserHelper.cpp.

#### 9.4.2.3 [uint1\\_2\\_p\\_t TRADEMGEN::DemandParserHelper::uint1\\_2\\_p](#)

Up-to-2-digit-integer parser

Definition at line 503 of file DemandParserHelper.cpp.

#### 9.4.2.4 [uint1\\_3\\_p\\_t TRADEMGEN::DemandParserHelper::uint1\\_3\\_p](#)

Up-to-3-digit-integer parser

Definition at line 506 of file DemandParserHelper.cpp.

#### 9.4.2.5 [uint4\\_p\\_t TRADEMGEN::DemandParserHelper::uint4\\_p](#)

4-digit-integer parser

Definition at line 509 of file DemandParserHelper.cpp.

#### 9.4.2.6 [uint1\\_4\\_p\\_t TRADEMGEN::DemandParserHelper::uint1\\_4\\_p](#)

Up-to-4-digit-integer parser

Definition at line 512 of file DemandParserHelper.cpp.

#### 9.4.2.7 [int1\\_p\\_t TRADEMGEN::DemandParserHelper::family\\_code\\_p](#)

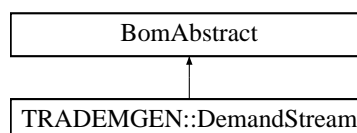
Family code parser

Definition at line 554 of file DemandParserHelper.cpp.

## 10 TraDemGen Class Documentation

### 10.1 BomAbstract Class Reference

Inheritance diagram for BomAbstract::



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

- trademgen/bom/[DemandStream.hpp](#)

## 10.2 TRADEMGEN::BomDisplay Class Reference

Utility class to display TraDemGen objects with a pretty format.

```
#include <trademgen/bom/BomDisplay.hpp>
```

### Static Public Member Functions

- static std::string [csvDisplay](#) (const SEVMGR::SEVMGR\_ServicePtr\_T)
- static void [csvDisplay](#) (std::ostream &, const [DemandStream](#) &)

### 10.2.1 Detailed Description

Utility class to display TraDemGen objects with a pretty format.

Definition at line 23 of file BomDisplay.hpp.

### 10.2.2 Member Function Documentation

#### 10.2.2.1 std::string TRADEMGEN::BomDisplay::csvDisplay (const SEVMGR::SEVMGR\_ServicePtr\_T) [static]

Recursively display (dump in the underlying output log stream) the objects of the BOM tree.

#### Parameters:

*std::ostream&* Output stream in which the BOM tree should be logged/dumped.

*const SEVMGR::SEVMGR\_ServicePtr\_T* Pointer on the SEVMgr service handler to display the queue key and to obtain the demand stream list.

Definition at line 45 of file BomDisplay.cpp.

Referenced by TRADEMGEN::TRADEMGEN\_Service::csvDisplay().

#### 10.2.2.2 void TRADEMGEN::BomDisplay::csvDisplay (std::ostream &, const [DemandStream](#) &) [static]

Recursively display (dump in the underlying output log stream) the objects of the BOM tree.

#### Parameters:

*std::ostream&* Output stream in which the BOM tree should be logged/dumped.

*const [DemandStream](#)&* Root of the BOM tree to be displayed.

Definition at line 87 of file BomDisplay.cpp.

References TRADEMGEN::DemandStream::display().

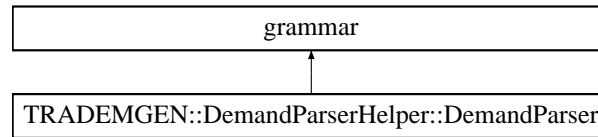
The documentation for this class was generated from the following files:

- trademgen/bom/[BomDisplay.hpp](#)
- trademgen/bom/[BomDisplay.cpp](#)



## 10.3 grammar Class Reference

Inheritance diagram for grammar::



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

- trademgen/command/[DemandParserHelper.hpp](#)

## 10.4 stdair::CategoricalAttribute< T > Struct Template Reference

Class modeling the distribution of values that can be taken by a categorical attribute.

```
#include <trademgen/basic/CategoricalAttribute.hpp>
```

### Public Types

- typedef std::map< T, [DictionaryKey\\_T](#) > [ProbabilityMassFunction\\_T](#)
- typedef std::map< [DictionaryKey\\_T](#), T > [InverseCumulativeDistribution\\_T](#)

### Public Member Functions

- const T & [getValue](#) (const [Probability\\_T](#) &iCumulativeProbability) const
- const std::string [displayProbabilityMassFunction](#) () const
- const std::string [displayInverseCumulativeDistribution](#) () const
- [CategoricalAttribute](#) (const [ProbabilityMassFunction\\_T](#) &iProbabilityMassFunction)
- [CategoricalAttribute](#) ()
- [CategoricalAttribute](#) (const [CategoricalAttribute](#) &iCategoricalAttribute)
- virtual [~CategoricalAttribute](#) ()
- void [determineInverseCumulativeDistributionFromProbabilityMassFunction](#) ()

### 10.4.1 Detailed Description

```
template<typename T> struct stdair::CategoricalAttribute< T >
```

Class modeling the distribution of values that can be taken by a categorical attribute.

Definition at line 21 of file CategoricalAttribute.hpp.

### 10.4.2 Member Typedef Documentation

#### 10.4.2.1 template<typename T> typedef std::map<T, [DictionaryKey\\_T](#)> [stdair::CategoricalAttribute< T >::ProbabilityMassFunction\\_T](#)

Define the probability mass function type.

Definition at line 28 of file CategoricalAttribute.hpp.

#### 10.4.2.2 `template<typename T> typedef std::map<DictionaryKey_T, T> stdair::CategoricalAttribute< T >::InverseCumulativeDistribution_T`

Define the inverse cumulative distribution type.

Definition at line 33 of file CategoricalAttribute.hpp.

### 10.4.3 Constructor & Destructor Documentation

#### 10.4.3.1 `template<typename T> stdair::CategoricalAttribute< T >::CategoricalAttribute (const ProbabilityMassFunction_T & iProbabilityMassFunction) [inline]`

Main constructor.

Definition at line 129 of file CategoricalAttribute.hpp.

References `stdair::CategoricalAttribute< T >::determineInverseCumulativeDistributionFromProbabilityMassFunction()`.

#### 10.4.3.2 `template<typename T> stdair::CategoricalAttribute< T >::CategoricalAttribute () [inline]`

Default constructor.

Definition at line 137 of file CategoricalAttribute.hpp.

#### 10.4.3.3 `template<typename T> stdair::CategoricalAttribute< T >::CategoricalAttribute (const CategoricalAttribute< T > & iCategoricalAttribute) [inline]`

Copy constructor.

Definition at line 142 of file CategoricalAttribute.hpp.

References `stdair::CategoricalAttribute< T >::determineInverseCumulativeDistributionFromProbabilityMassFunction()`.

#### 10.4.3.4 `template<typename T> virtual stdair::CategoricalAttribute< T >::~~CategoricalAttribute () [inline, virtual]`

Destructor.

Definition at line 150 of file CategoricalAttribute.hpp.

### 10.4.4 Member Function Documentation

#### 10.4.4.1 `template<typename T> const T& stdair::CategoricalAttribute< T >::getValue (const Probability_T & iCumulativeProbability) const [inline]`

Get value from inverse cumulative distribution.

Definition at line 67 of file CategoricalAttribute.hpp.

References `stdair::CategoricalAttribute< T >::displayInverseCumulativeDistribution()`.

#### 10.4.4.2 `template<typename T> const std::string stdair::CategoricalAttribute< T >::displayProbabilityMassFunction () const [inline]`

Display probability mass function.

Definition at line 91 of file CategoricalAttribute.hpp.

**10.4.4.3** `template<typename T> const std::string stdair::CategoricalAttribute< T >::displayInverseCumulativeDistribution () const [inline]`

Display inverse cumulative distribution.

Definition at line 111 of file CategoricalAttribute.hpp.

Referenced by `stdair::CategoricalAttribute< T >::getValue()`.

**10.4.4.4** `template<typename T> void stdair::CategoricalAttribute< T >::determineInverseCumulativeDistributionFromProbabilityMassFunction () [inline]`

Determine inverse cumulative distribution from probability mass function (initialisation).

Definition at line 157 of file CategoricalAttribute.hpp.

Referenced by `stdair::CategoricalAttribute< T >::CategoricalAttribute()`.

The documentation for this struct was generated from the following file:

- [trademgen/basic/CategoricalAttribute.hpp](#)

## 10.5 TRADEMGEN::CategoricalAttributeLite< T > Struct Template Reference

Class modeling the distribution of values that can be taken by a categorical attribute.

```
#include <trademgen/basic/CategoricalAttributeLite.hpp>
```

### Public Types

- `typedef std::map< T, stdair::Probability_T > ProbabilityMassFunction_T`

### Public Member Functions

- `const T & getValue (const stdair::Probability_T &iCumulativeProbability) const`
- `bool checkValue (const T &iValue) const`
- `const std::string displayProbabilityMass () const`
- `CategoricalAttributeLite (const ProbabilityMassFunction_T &iValueMap)`
- `CategoricalAttributeLite ()`
- `CategoricalAttributeLite (const CategoricalAttributeLite &iCAL)`
- `CategoricalAttributeLite & operator= (const CategoricalAttributeLite &iCAL)`
- `virtual ~CategoricalAttributeLite ()`

### 10.5.1 Detailed Description

`template<typename T> struct TRADEMGEN::CategoricalAttributeLite< T >`

Class modeling the distribution of values that can be taken by a categorical attribute.

Definition at line 27 of file CategoricalAttributeLite.hpp.

## 10.5.2 Member Typedef Documentation

**10.5.2.1** `template<typename T> typedef std::map<T, stdair::Probability_T> TRADEMGEN::CategoricalAttributeLite< T >::ProbabilityMassFunction_T`

Type for the probability mass function.

Definition at line 33 of file CategoricalAttributeLite.hpp.

## 10.5.3 Constructor & Destructor Documentation

**10.5.3.1** `template<typename T> TRADEMGEN::CategoricalAttributeLite< T >::CategoricalAttributeLite(const ProbabilityMassFunction_T & iValueMap) [inline]`

Main constructor.

Definition at line 95 of file CategoricalAttributeLite.hpp.

**10.5.3.2** `template<typename T> TRADEMGEN::CategoricalAttributeLite< T >::CategoricalAttributeLite() [inline]`

Default constructor.

Definition at line 103 of file CategoricalAttributeLite.hpp.

**10.5.3.3** `template<typename T> TRADEMGEN::CategoricalAttributeLite< T >::CategoricalAttributeLite(const CategoricalAttributeLite< T > & iCAL) [inline]`

Copy constructor.

Definition at line 109 of file CategoricalAttributeLite.hpp.

**10.5.3.4** `template<typename T> virtual TRADEMGEN::CategoricalAttributeLite< T >::~~CategoricalAttributeLite() [inline, virtual]`

Destructor.

Definition at line 128 of file CategoricalAttributeLite.hpp.

## 10.5.4 Member Function Documentation

**10.5.4.1** `template<typename T> const T& TRADEMGEN::CategoricalAttributeLite< T >::getValue(const stdair::Probability_T & iCumulativeProbability) const [inline]`

Get value from inverse cumulative distribution.

Definition at line 41 of file CategoricalAttributeLite.hpp.

References `TRADEMGENT::CategoricalAttributeLite< T >::displayProbabilityMass()`, and `TRADEMGENT::DictionaryManager::valueToKey()`.

Referenced by `TRADEMGENT::DemandStream::generateChannel()`, `TRADEMGENT::DemandStream::generateFrequentFlyer()`, `TRADEMGENT::DemandStream::generateStayDuration()`, `TRADEMGENT::DemandStream::generateTripType()`, and `TRADEMGENT::DemandCharacteristics::getPOSValue()`.

**10.5.4.2** `template<typename T> bool TRADEMGEN::CategoricalAttributeLite< T >::checkValue(const T & iValue) const` [inline]

Check if a value belongs to the value list.

Definition at line 61 of file CategoricalAttributeLite.hpp.

Referenced by TRADEMGEN::DemandCharacteristics::checkPOSValue().

**10.5.4.3** `template<typename T> const std::string TRADEMGEN::CategoricalAttributeLite< T >::displayProbabilityMass() const` [inline]

Display probability mass function.

Definition at line 76 of file CategoricalAttributeLite.hpp.

References TRADEMGEN::DictionaryManager::keyToValue().

Referenced by TRADEMGEN::DemandCharacteristics::describe(), TRADEMGEN::DemandStream::display(), and TRADEMGEN::CategoricalAttributeLite< T >::getValue().

**10.5.4.4** `template<typename T> CategoricalAttributeLite& TRADEMGEN::CategoricalAttributeLite< T >::operator= (const CategoricalAttributeLite< T > & iCAL)` [inline]

Copy operator.

Definition at line 118 of file CategoricalAttributeLite.hpp.

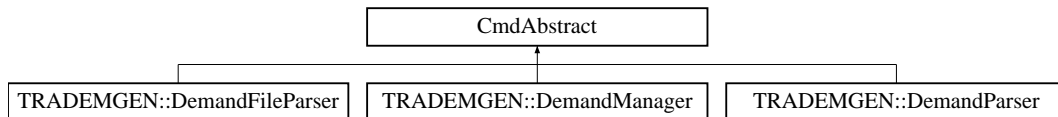
References TRADEMGEN::CategoricalAttributeLite< T >::\_cumulativeDistribution, TRADEMGEN::CategoricalAttributeLite< T >::\_size, and TRADEMGEN::CategoricalAttributeLite< T >::\_valueArray.

The documentation for this struct was generated from the following file:

- trademgen/basic/[CategoricalAttributeLite.hpp](#)

## 10.6 CmdAbstract Class Reference

Inheritance diagram for CmdAbstract::



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

- trademgen/command/[DemandParser.hpp](#)
- trademgen/command/[DemandManager.hpp](#)
- trademgen/command/[DemandParserHelper.hpp](#)

## 10.7 TRADEMGEN::ContinuousAttribute< T > Struct Template Reference

```
#include <trademgen/basic/ContinuousAttribute.hpp>
```

## Public Types

- typedef std::multimap< T, [DictionaryKey\\_T](#) > [ContinuousDistribution\\_T](#)
- typedef std::multimap< [DictionaryKey\\_T](#), T > [ContinuousInverseDistribution\\_T](#)

## Public Member Functions

- const T [getValue](#) (const stdair::Probability\_T &iCumulativeProbability) const
- const std::string [displayCumulativeDistribution](#) () const
- const std::string [displayInverseCumulativeDistribution](#) () const
- [ContinuousAttribute](#) ()
- [ContinuousAttribute](#) (const [ContinuousDistribution\\_T](#) &iCumulativeDistribution)
- [ContinuousAttribute](#) (const [ContinuousAttribute](#) &iContinuousAttribute)
- virtual [~ContinuousAttribute](#) ()
- void [determineInverseCumulativeDistributionFromCumulativeDistribution](#) ()

### 10.7.1 Detailed Description

**template<class T> struct TRADEMGEN::ContinuousAttribute< T >**

Class modeling the distribution of values that can be taken by a continuous attribute.

Definition at line 21 of file ContinuousAttribute.hpp.

### 10.7.2 Member Typedef Documentation

**10.7.2.1** **template<class T> typedef std::multimap<T, [DictionaryKey\\_T](#)> [TRADEMGEN::ContinuousAttribute< T >::ContinuousDistribution\\_T](#)**

Definition at line 26 of file ContinuousAttribute.hpp.

**10.7.2.2** **template<class T> typedef std::multimap<[DictionaryKey\\_T](#), T> [TRADEMGEN::ContinuousAttribute< T >::ContinuousInverseDistribution\\_T](#)**

Definition at line 27 of file ContinuousAttribute.hpp.

### 10.7.3 Constructor & Destructor Documentation

**10.7.3.1** **template<class T> [TRADEMGEN::ContinuousAttribute< T >::ContinuousAttribute](#) ()**  
[inline]

Constructor by default

Definition at line 113 of file ContinuousAttribute.hpp.

**10.7.3.2** **template<class T> [TRADEMGEN::ContinuousAttribute< T >::ContinuousAttribute](#) (const [ContinuousDistribution\\_T](#) &iCumulativeDistribution)** [inline]

Constructor

Definition at line 116 of file ContinuousAttribute.hpp.

References [TRADEMGEN::ContinuousAttribute< T >::determineInverseCumulativeDistributionFromCumulativeDistribution\(\)](#).

**10.7.3.3** `template<class T> TRADEMGEN::ContinuousAttribute< T >::ContinuousAttribute(const ContinuousAttribute< T > & iContinuousAttribute) [inline]`

Copy constructor

Definition at line 122 of file ContinuousAttribute.hpp.

**10.7.3.4** `template<class T> virtual TRADEMGEN::ContinuousAttribute< T >::~~ContinuousAttribute() [inline, virtual]`

Destructor

Definition at line 128 of file ContinuousAttribute.hpp.

## 10.7.4 Member Function Documentation

**10.7.4.1** `template<class T> const T TRADEMGEN::ContinuousAttribute< T >::getValue(const stdair::Probability_T & iCumulativeProbability) const [inline]`

Get value from inverse cumulative distribution.

Definition at line 52 of file ContinuousAttribute.hpp.

References TRADEMGEN::DictionaryManager::keyToValue(), and TRADEMGEN::DictionaryManager::valueToKey().

**10.7.4.2** `template<class T> const std::string TRADEMGEN::ContinuousAttribute< T >::displayCumulativeDistribution() const [inline]`

Display cumulative distribution

Definition at line 83 of file ContinuousAttribute.hpp.

References TRADEMGEN::DictionaryManager::keyToValue().

**10.7.4.3** `template<class T> const std::string TRADEMGEN::ContinuousAttribute< T >::displayInverseCumulativeDistribution() const [inline]`

Display inverse cumulative distribution

Definition at line 99 of file ContinuousAttribute.hpp.

References TRADEMGEN::DictionaryManager::keyToValue().

**10.7.4.4** `template<class T> void TRADEMGEN::ContinuousAttribute< T >::determineInverseCumulativeDistributionFromCumulativeDistribution() [inline]`

Determine inverse cumulative distribution from cumulative distribution (initialisation).

Definition at line 132 of file ContinuousAttribute.hpp.

Referenced by TRADEMGEN::ContinuousAttribute< T >::ContinuousAttribute().

The documentation for this struct was generated from the following file:

- trademgen/basic/[ContinuousAttribute.hpp](#)

## 10.8 TRADEMGEN::ContinuousAttributeLite< T > Struct Template Reference

Class modeling the distribution of values that can be taken by a continuous attribute.

```
#include <trademgen/basic/ContinuousAttributeLite.hpp>
```

### Public Types

- typedef std::map< T, stdair::Probability\_T > [ContinuousDistribution\\_T](#)

### Public Member Functions

- const T [getValue](#) (const stdair::Probability\_T &iCumulativeProbability) const
- const double [getDerivativeValue](#) (const T iKey) const
- const T [getUpperBound](#) (const T iKey) const
- const std::string [displayCumulativeDistribution](#) () const
- [ContinuousAttributeLite](#) (const [ContinuousDistribution\\_T](#) &iValueMap)
- [ContinuousAttributeLite](#) (const [ContinuousAttributeLite](#) &iCAL)
- [ContinuousAttributeLite](#) & [operator=](#) (const [ContinuousAttributeLite](#) &iCAL)
- virtual [~ContinuousAttributeLite](#) ()

### 10.8.1 Detailed Description

```
template<typename T> struct TRADEMGEN::ContinuousAttributeLite< T >
```

Class modeling the distribution of values that can be taken by a continuous attribute.

Definition at line 26 of file ContinuousAttributeLite.hpp.

### 10.8.2 Member Typedef Documentation

**10.8.2.1** template<typename T> typedef std::map<T, stdair::Probability\_T> [TRADEMGEN::ContinuousAttributeLite< T >::ContinuousDistribution\\_T](#)

Type for the probability mass function.

Definition at line 32 of file ContinuousAttributeLite.hpp.

### 10.8.3 Constructor & Destructor Documentation

**10.8.3.1** template<typename T> [TRADEMGEN::ContinuousAttributeLite< T >::ContinuousAttributeLite](#) (const [ContinuousDistribution\\_T](#) &iValueMap) [inline]

Constructor.

Definition at line 157 of file ContinuousAttributeLite.hpp.

**10.8.3.2** template<typename T> [TRADEMGEN::ContinuousAttributeLite< T >::ContinuousAttributeLite](#) (const [ContinuousAttributeLite< T >](#) &iCAL) [inline]

Copy constructor.

Definition at line 165 of file ContinuousAttributeLite.hpp.



**10.8.3.3** `template<typename T> virtual TRADEMGEN::ContinuousAttributeLite< T >::~~ContinuousAttributeLite () [inline, virtual]`

Destructor.

Definition at line 184 of file ContinuousAttributeLite.hpp.

## 10.8.4 Member Function Documentation

**10.8.4.1** `template<typename T> const T TRADEMGEN::ContinuousAttributeLite< T >::getValue (const stdair::Probability_T & iCumulativeProbability) const [inline]`

Get value from inverse cumulative distribution.

Definition at line 39 of file ContinuousAttributeLite.hpp.

References TRADEMGEN::DictionaryManager::keyToValue(), and TRADEMGEN::DictionaryManager::valueToKey().

Referenced by TRADEMGEN::DemandStream::generateTimeOfRequestPoissonProcess(), TRADEMGEN::DemandStream::generateTimeOfRequestStatisticsOrder(), TRADEMGEN::DemandStream::generateValueOfTime(), and TRADEMGEN::DemandStream::generateWTP().

**10.8.4.2** `template<typename T> const double TRADEMGEN::ContinuousAttributeLite< T >::getDerivativeValue (const T iKey) const [inline]`

Get the value of the derivative function in a key point.

Definition at line 82 of file ContinuousAttributeLite.hpp.

References TRADEMGEN::DictionaryManager::keyToValue().

Referenced by TRADEMGEN::DemandStream::generateTimeOfRequestPoissonProcess().

**10.8.4.3** `template<typename T> const T TRADEMGEN::ContinuousAttributeLite< T >::getUpperBound (const T iKey) const [inline]`

Get the upper bound.

Definition at line 116 of file ContinuousAttributeLite.hpp.

Referenced by TRADEMGEN::DemandStream::generateTimeOfRequestPoissonProcess().

**10.8.4.4** `template<typename T> const std::string TRADEMGEN::ContinuousAttributeLite< T >::displayCumulativeDistribution () const [inline]`

Display cumulative distribution.

Definition at line 135 of file ContinuousAttributeLite.hpp.

References TRADEMGEN::DictionaryManager::keyToValue().

Referenced by TRADEMGEN::DemandCharacteristics::describe().

**10.8.4.5** `template<typename T> ContinuousAttributeLite& TRADEMGEN::ContinuousAttributeLite< T >::operator= (const ContinuousAttributeLite< T > & iCAL) [inline]`

Copy operator.

Definition at line 174 of file ContinuousAttributeLite.hpp.

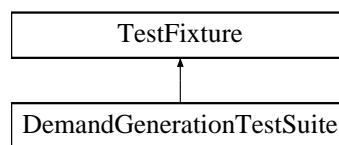
References TRADEMGEN::ContinuousAttributeLite< T >::\_cumulativeDistribution, TRADEMGEN::ContinuousAttributeLite< T >::\_size, and TRADEMGEN::ContinuousAttributeLite< T >::\_valueArray.

The documentation for this struct was generated from the following file:

- trademgen/basic/[ContinuousAttributeLite.hpp](#)

## 10.9 TestFixture Class Reference

Inheritance diagram for TestFixture::



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

- test/trademgen/[DemandGenerationTestSuite.hpp](#)

## 10.10 TRADEMGEN::DBManager Class Reference

```
#include <trademgen/command/DBManager.hpp>
```

### Static Public Member Functions

- static void [updateAirlineInDB](#) (stdair::DBSession\_T &, const stdair::AirlineStruct &)
- static bool [retrieveAirline](#) (stdair::DBSession\_T &, const stdair::AirlineCode\_T &, stdair::AirlineStruct &)
- static void [prepareSelectStatement](#) (stdair::DBSession\_T &, stdair::DBRequestStatement\_T &, stdair::AirlineStruct &)
- static bool [iterateOnStatement](#) (stdair::DBRequestStatement\_T &, stdair::AirlineStruct &, const bool iShouldDoReset)

### 10.10.1 Detailed Description

Class building the Business Object Model (BOM) from data retrieved from the database.

Definition at line 20 of file DBManager.hpp.

### 10.10.2 Member Function Documentation

**10.10.2.1 void TRADEMGEN::DBManager::updateAirlineInDB (stdair::DBSession\_T &, const stdair::AirlineStruct &) [static]**

Update the fields of the database row corresponding to the given BOM object.

Definition at line 121 of file DBManager.cpp.

**10.10.2.2** `bool TRADEMGEN::DBManager::retrieveAirline (stdair::DBSession_T &, const stdair::AirlineCode_T &, stdair::AirlineStruct &) [static]`

Retrieve, from the (MySQL) database, the row corresponding to the given BOM code, and fill the given BOM object with that retrieved data.

Definition at line 157 of file DBManager.cpp.

References `iterateOnStatement()`.

**10.10.2.3** `void TRADEMGEN::DBManager::prepareSelectStatement (stdair::DBSession_T &, stdair::DBRequestStatement_T &, stdair::AirlineStruct &) [static]`

Prepare (parse and put in cache) the SQL statement.

Definition at line 24 of file DBManager.cpp.

**10.10.2.4** `bool TRADEMGEN::DBManager::iterateOnStatement (stdair::DBRequestStatement_T &, stdair::AirlineStruct &, const bool iShouldDoReset) [static]`

Iterate on the SQL statement.

The SQL has to be already prepared. `const bool` Tells whether the Airline object should be reset.

Definition at line 97 of file DBManager.cpp.

Referenced by `retrieveAirline()`.

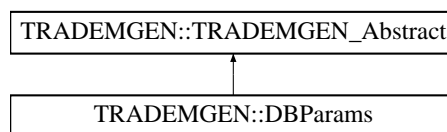
The documentation for this class was generated from the following files:

- [trademgen/command/DBManager.hpp](#)
- [trademgen/command/DBManager.cpp](#)

## 10.11 TRADEMGEN::DBParams Struct Reference

```
#include <trademgen/DBParams.hpp>
```

Inheritance diagram for TRADEMGEN::DBParams::



### Public Member Functions

- `std::string` [getUser](#) () const
- `std::string` [getPassword](#) () const
- `std::string` [getHost](#) () const
- `std::string` [getPort](#) () const
- `std::string` [getDBName](#) () const
- `void` [setUser](#) (const `std::string` &`iUser`)
- `void` [setPassword](#) (const `std::string` &`iPasswd`)
- `void` [setHost](#) (const `std::string` &`iHost`)

- void [setPort](#) (const std::string &iPort)
- void [setDBName](#) (const std::string &iDBName)
- bool [check](#) () const
- void [toStream](#) (std::ostream &ioOut) const
- void [fromStream](#) (std::istream &)
- std::string [toShortString](#) () const
- std::string [toString](#) () const
- [DBParams](#) (const std::string &iDBUser, const std::string &iDBPasswd, const std::string &iDBHost, const std::string &iDBPort, const std::string &iDBName)
- virtual [~DBParams](#) ()

### 10.11.1 Detailed Description

Structure modelling a (geographical) dbparams.

Definition at line 21 of file DBParams.hpp.

### 10.11.2 Constructor & Destructor Documentation

**10.11.2.1 TRADEMGEN::DBParams::DBParams (const std::string &iDBUser, const std::string &iDBPasswd, const std::string &iDBHost, const std::string &iDBPort, const std::string &iDBName) [inline]**

Main Constructor.

Definition at line 119 of file DBParams.hpp.

**10.11.2.2 virtual TRADEMGEN::DBParams::~~DBParams () [inline, virtual]**

Destructor.

Definition at line 132 of file DBParams.hpp.

### 10.11.3 Member Function Documentation

**10.11.3.1 std::string TRADEMGEN::DBParams::getUser () const [inline]**

Get the database user name.

Definition at line 25 of file DBParams.hpp.

**10.11.3.2 std::string TRADEMGEN::DBParams::getPassword () const [inline]**

Get the database user password.

Definition at line 30 of file DBParams.hpp.

**10.11.3.3 std::string TRADEMGEN::DBParams::getHost () const [inline]**

Get the database host name.

Definition at line 35 of file DBParams.hpp.

**10.11.3.4** `std::string TRADEMGEN::DBParams::getPort () const` `[inline]`

Get the database port number.

Definition at line 40 of file DBParams.hpp.

**10.11.3.5** `std::string TRADEMGEN::DBParams::getDBName () const` `[inline]`

Get the database name.

Definition at line 45 of file DBParams.hpp.

**10.11.3.6** `void TRADEMGEN::DBParams::setUser (const std::string & iUser)` `[inline]`

Set the database user name.

Definition at line 52 of file DBParams.hpp.

**10.11.3.7** `void TRADEMGEN::DBParams::setPassword (const std::string & iPasswd)`  
`[inline]`

Set the database password.

Definition at line 57 of file DBParams.hpp.

**10.11.3.8** `void TRADEMGEN::DBParams::setHost (const std::string & iHost)` `[inline]`

Set the database host name.

Definition at line 62 of file DBParams.hpp.

**10.11.3.9** `void TRADEMGEN::DBParams::setPort (const std::string & iPort)` `[inline]`

Set the database port number.

Definition at line 67 of file DBParams.hpp.

**10.11.3.10** `void TRADEMGEN::DBParams::setDBName (const std::string & iDBName)`  
`[inline]`

Set the database name.

Definition at line 72 of file DBParams.hpp.

**10.11.3.11** `bool TRADEMGEN::DBParams::check () const` `[inline]`

Check that all the parameters are fine.

Definition at line 80 of file DBParams.hpp.

**10.11.3.12** `void TRADEMGEN::DBParams::toStream (std::ostream & ioOut) const` `[inline, virtual]`

Dump a structure into an output stream.

**Parameters:**

*ostream&* the output stream.

Implements [TRADEMGEN::TRADEMGEN\\_Abstract](#).

Definition at line 93 of file DBParams.hpp.

References toString().

**10.11.3.13** `void TRADEMGEN::DBParams::fromStream (std::istream &) [inline, virtual]`

Read a structure from an input stream.

**Parameters:**

*istream&* the input stream.

Implements [TRADEMGEN::TRADEMGEN\\_Abstract](#).

Definition at line 99 of file DBParams.hpp.

**10.11.3.14** `std::string TRADEMGEN::DBParams::toShortString () const [inline]`

Get a short display of the [DBParams](#) structure.

Definition at line 103 of file DBParams.hpp.

**10.11.3.15** `std::string TRADEMGEN::DBParams::toString () const [inline, virtual]`

Get the serialised version of the [DBParams](#) structure.

Implements [TRADEMGEN::TRADEMGEN\\_Abstract](#).

Definition at line 110 of file DBParams.hpp.

Referenced by toStream().

The documentation for this struct was generated from the following file:

- [trademgen/DBParams.hpp](#)

## 10.12 TRADEMGEN::DefaultMap Struct Reference

```
#include <trademgen/basic/BasConst_DemandGeneration.hpp>
```

### Static Public Member Functions

- static [POSProbabilityMassFunction\\_T createPOSProbMass \(\)](#)
- static [FRAT5Pattern\\_T createFRAT5Pattern \(\)](#)

#### 10.12.1 Detailed Description

Default PoS probability mass.

Definition at line 24 of file BasConst\_DemandGeneration.hpp.

### 10.12.2 Member Function Documentation

#### 10.12.2.1 [POSProbabilityMassFunction\\_T](#) TRADEMGEN::DefaultMap::createPOSProbMass () [static]

Default PoS probability mass.

Definition at line 20 of file BasConst.cpp.

#### 10.12.2.2 [FRAT5Pattern\\_T](#) TRADEMGEN::DefaultMap::createFRAT5Pattern () [static]

Default FRAT5 pattern.

Definition at line 41 of file BasConst.cpp.

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

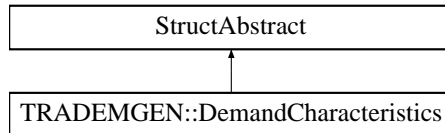
- [trademgen/basic/BasConst\\_DemandGeneration.hpp](#)
- [trademgen/basic/BasConst.cpp](#)

## 10.13 TRADEMGEN::DemandCharacteristics Struct Reference

Class modeling the characteristics of a demand type.

```
#include <trademgen/basic/DemandCharacteristics.hpp>
```

Inheritance diagram for TRADEMGEN::DemandCharacteristics::



### Public Member Functions

- const stdair::AirportCode\_T & [getPOSValue](#) (const stdair::Probability\_T &iCumulativeProbability) const
- bool [checkPOSValue](#) (const stdair::AirportCode\_T &iPOS) const
- const std::string [describe](#) () const
- [DemandCharacteristics](#) (const [ArrivalPatternCumulativeDistribution\\_T](#) &, const [POSProbabilityMassFunction\\_T](#) &, const [ChannelProbabilityMassFunction\\_T](#) &, const [TripTypeProbabilityMassFunction\\_T](#) &, const [StayDurationProbabilityMassFunction\\_T](#) &, const [FrequentFlyerProbabilityMassFunction\\_T](#) &, const stdair::ChangeFeesRatio\_T &, const stdair::Disutility\_T &, const stdair::NonRefundableRatio\_T &, const stdair::Disutility\_T &, const [PreferredDepartureTimeContinuousDistribution\\_T](#) &, const stdair::WTP\_T &, const [ValueOfTimeContinuousDistribution\\_T](#) &)
- [DemandCharacteristics](#) ()
- [DemandCharacteristics](#) (const [DemandCharacteristics](#) &)
- [~DemandCharacteristics](#) ()

**Public Attributes**

- [ContinuousFloatDuration\\_T \\_arrivalPattern](#)
- [POSProbabilityMass\\_T \\_posProbabilityMass](#)
- [ChannelProbabilityMass\\_T \\_channelProbabilityMass](#)
- [TripTypeProbabilityMass\\_T \\_tripTypeProbabilityMass](#)
- [StayDurationProbabilityMass\\_T \\_stayDurationProbabilityMass](#)
- [FrequentFlyerProbabilityMass\\_T \\_frequentFlyerProbabilityMass](#)
- [stdair::ChangeFeesRatio\\_T \\_changeFeeProb](#)
- [stdair::Disutility\\_T \\_changeFeeDisutility](#)
- [stdair::NonRefundableRatio\\_T \\_nonRefundableProb](#)
- [stdair::Disutility\\_T \\_nonRefundableDisutility](#)
- [PreferredDepartureTimeCumulativeDistribution\\_T](#) [\\_preferredDepartureTimeCumulativeDistribution](#)
- [stdair::WTP\\_T \\_minWTP](#)
- [CumulativeDistribution\\_T \\_frat5Pattern](#)
- [ValueOfTimeCumulativeDistribution\\_T \\_valueOfTimeCumulativeDistribution](#)

**10.13.1 Detailed Description**

Class modeling the characteristics of a demand type.

Definition at line 21 of file DemandCharacteristics.hpp.

**10.13.2 Constructor & Destructor Documentation**

**10.13.2.1 TRADEMGEN::DemandCharacteristics::DemandCharacteristics** (const [ArrivalPatternCumulativeDistribution\\_T](#) &, const [POSProbabilityMassFunction\\_T](#) &, const [ChannelProbabilityMassFunction\\_T](#) &, const [TripTypeProbabilityMassFunction\\_T](#) &, const [StayDurationProbabilityMassFunction\\_T](#) &, const [FrequentFlyerProbabilityMassFunction\\_T](#) &, const [stdair::ChangeFeesRatio\\_T](#) &, const [stdair::Disutility\\_T](#) &, const [stdair::NonRefundableRatio\\_T](#) &, const [stdair::Disutility\\_T](#) &, const [PreferredDepartureTimeContinuousDistribution\\_T](#) &, const [stdair::WTP\\_T](#) &, const [ValueOfTimeContinuousDistribution\\_T](#) &)

Constructor.

Definition at line 50 of file DemandCharacteristics.cpp.

**10.13.2.2 TRADEMGEN::DemandCharacteristics::DemandCharacteristics ()**

Default constructor.

Definition at line 16 of file DemandCharacteristics.cpp.

**10.13.2.3 TRADEMGEN::DemandCharacteristics::DemandCharacteristics** (const [DemandCharacteristics](#) &)

Copy constructor.

Definition at line 32 of file DemandCharacteristics.cpp.



**10.13.2.4 TRADEMGEN::DemandCharacteristics::~~DemandCharacteristics ()**

Destructor.

Definition at line 79 of file DemandCharacteristics.cpp.

**10.13.3 Member Function Documentation****10.13.3.1 const stdair::AirportCode\_T & TRADEMGEN::DemandCharacteristics::getPOSValue (const stdair::Probability\_T & iCumulativeProbability) const**

Get the POS corresponding to the cumulative probability

Definition at line 84 of file DemandCharacteristics.cpp.

References `_posProbabilityMass`, and `TRADEMGENT::CategoricalAttributeLite< T >::getValue()`.

Referenced by `TRADEMGENT::DemandStream::generatePOS()`.

**10.13.3.2 bool TRADEMGEN::DemandCharacteristics::checkPOSValue (const stdair::AirportCode\_T & iPOS) const**

Check that the POS is within the distribution.

Definition at line 90 of file DemandCharacteristics.cpp.

References `_posProbabilityMass`, and `TRADEMGENT::CategoricalAttributeLite< T >::checkValue()`.

**10.13.3.3 const std::string TRADEMGEN::DemandCharacteristics::describe () const**

Give a description of the structure (for display purposes).

Definition at line 95 of file DemandCharacteristics.cpp.

References `_arrivalPattern`, `_changeFeeDisutility`, `_changeFeeProb`, `_channelProbabilityMass`, `_frequentFlyerProbabilityMass`, `_minWTP`, `_nonRefundableDisutility`, `_nonRefundableProb`, `_posProbabilityMass`, `_preferredDepartureTimeCumulativeDistribution`, `_stayDurationProbabilityMass`, `_tripTypeProbabilityMass`, `_valueOfTimeCumulativeDistribution`, `TRADEMGENT::ContinuousAttributeLite< T >::displayCumulativeDistribution()`, and `TRADEMGENT::CategoricalAttributeLite< T >::displayProbabilityMass()`.

Referenced by `TRADEMGENT::DemandStream::display()`.

**10.13.4 Member Data Documentation****10.13.4.1 ContinuousFloatDuration\_T TRADEMGEN::DemandCharacteristics::\_arrivalPattern**

Arrival pattern (cumulative distribution of timing of arrival of requests (negative number of days between departure date and request date).

Definition at line 87 of file DemandCharacteristics.hpp.

Referenced by `describe()`, `TRADEMGENT::DemandStream::generateTimeOfRequestPoissonProcess()`, and `TRADEMGENT::DemandStream::generateTimeOfRequestStatisticsOrder()`.

**10.13.4.2 POSProbabilityMass\_T TRADEMGEN::DemandCharacteristics::\_posProbabilityMass**

POS probability mass.

Definition at line 92 of file DemandCharacteristics.hpp.

Referenced by checkPOSValue(), describe(), and getPOSValue().

#### 10.13.4.3 ChannelProbabilityMass\_T TRADEMGEN::DemandCharacteristics::\_channel-ProbabilityMass

Channel probability mass.

Definition at line 97 of file DemandCharacteristics.hpp.

Referenced by describe(), and TRADEMGEN::DemandStream::generateChannel().

#### 10.13.4.4 TripTypeProbabilityMass\_T TRADEMGEN::DemandCharacteristics::\_tripType-ProbabilityMass

Trip type probability mass.

Definition at line 102 of file DemandCharacteristics.hpp.

Referenced by describe(), and TRADEMGEN::DemandStream::generateTripType().

#### 10.13.4.5 StayDurationProbabilityMass\_T TRADEMGEN::DemandCharacteristics::\_stay-DurationProbabilityMass

Stay duration probability mass.

Definition at line 107 of file DemandCharacteristics.hpp.

Referenced by describe(), and TRADEMGEN::DemandStream::generateStayDuration().

#### 10.13.4.6 FrequentFlyerProbabilityMass\_T TRADEMGEN::DemandCharacteristics::\_frequent-FlyerProbabilityMass

Frequent flyer probability mass.

Definition at line 112 of file DemandCharacteristics.hpp.

Referenced by describe(), and TRADEMGEN::DemandStream::generateFrequentFlyer().

#### 10.13.4.7 stdair::ChangeFeesRatio\_T TRADEMGEN::DemandCharacteristics::\_changeFeeProb

Change fee restriction acceptance probability.

Definition at line 117 of file DemandCharacteristics.hpp.

Referenced by describe(), and TRADEMGEN::DemandStream::generateChangeFees().

#### 10.13.4.8 stdair::Disutility\_T TRADEMGEN::DemandCharacteristics::\_changeFeeDisutility

Change fee disutility.

Definition at line 122 of file DemandCharacteristics.hpp.

Referenced by describe(), TRADEMGEN::DemandStream::generateNextRequest(), and TRADEMGEN::DemandStream::getChangeFeeDisutility().

**10.13.4.9** `stdair::NonRefundableRatio_T` [TRADEMGEN::DemandCharacteristics::\\_nonRefundableProb](#)

Non refundable restriction acceptance probability.

Definition at line 127 of file DemandCharacteristics.hpp.

Referenced by `describe()`, and `TRADEMGEN::DemandStream::generateNonRefundable()`.

**10.13.4.10** `stdair::Disutility_T` [TRADEMGEN::DemandCharacteristics::\\_nonRefundableDisutility](#)

Non refundable disutility.

Definition at line 132 of file DemandCharacteristics.hpp.

Referenced by `describe()`, `TRADEMGEN::DemandStream::generateNextRequest()`, and `TRADEMGEN::DemandStream::getNonRefundableDisutility()`.

**10.13.4.11** [PreferredDepartureTimeCumulativeDistribution\\_T](#) [TRADEMGEN::DemandCharacteristics::\\_preferredDepartureTimeCumulativeDistribution](#)

Preferred departure time cumulative distribution.

Definition at line 137 of file DemandCharacteristics.hpp.

Referenced by `describe()`.

**10.13.4.12** `stdair::WTP_T` [TRADEMGEN::DemandCharacteristics::\\_minWTP](#)

Min Willingness-to-pay, used for the computation of the WTP of each request.

Definition at line 143 of file DemandCharacteristics.hpp.

Referenced by `describe()`, and `TRADEMGEN::DemandStream::generateWTP()`.

**10.13.4.13** [CumulativeDistribution\\_T](#) [TRADEMGEN::DemandCharacteristics::\\_frat5Pattern](#)

FRAT5 pattern, used for the computation of WTP.

Definition at line 148 of file DemandCharacteristics.hpp.

Referenced by `TRADEMGEN::DemandStream::generateWTP()`.

**10.13.4.14** [ValueOfTimeCumulativeDistribution\\_T](#) [TRADEMGEN::DemandCharacteristics::\\_valueOfTimeCumulativeDistribution](#)

Value of time cumulative distribution.

Definition at line 153 of file DemandCharacteristics.hpp.

Referenced by `describe()`, and `TRADEMGEN::DemandStream::generateValueOfTime()`.

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

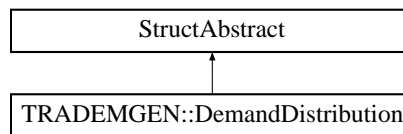
- `trademgen/basic/DemandCharacteristics.hpp`
- `trademgen/basic/DemandCharacteristics.cpp`

## 10.14 TRADEMGEN::DemandDistribution Struct Reference

Class modeling the distribution of a demand type.

```
#include <trademgen/basic/DemandDistribution.hpp>
```

Inheritance diagram for TRADEMGEN::DemandDistribution::



### Public Member Functions

- [DemandDistribution](#) (const stdair::NbOfRequests\_T &iMean, const stdair::StdDevValue\_T &iStdDev)
- [DemandDistribution](#) ()
- [DemandDistribution](#) (const [DemandDistribution](#) &)
- [~DemandDistribution](#) ()
- void [fromStream](#) (std::istream &ioIn)
- const std::string [describe](#) () const
- std::string [display](#) () const

### Public Attributes

- stdair::NbOfRequests\_T [\\_meanNumberOfRequests](#)
- stdair::StdDevValue\_T [\\_stdDevNumberOfRequests](#)

#### 10.14.1 Detailed Description

Class modeling the distribution of a demand type.

Definition at line 20 of file DemandDistribution.hpp.

#### 10.14.2 Constructor & Destructor Documentation

##### 10.14.2.1 TRADEMGEN::DemandDistribution::DemandDistribution (const stdair::NbOfRequests\_T &iMean, const stdair::StdDevValue\_T &iStdDev)

Constructor.

Definition at line 15 of file DemandDistribution.cpp.

##### 10.14.2.2 TRADEMGEN::DemandDistribution::DemandDistribution ()

Default constructor.

Definition at line 22 of file DemandDistribution.cpp.

**10.14.2.3 TRADEMGEN::DemandDistribution::DemandDistribution (const [DemandDistribution](#) &)**

Copy constructor.

Definition at line 31 of file DemandDistribution.cpp.

**10.14.2.4 TRADEMGEN::DemandDistribution::~~DemandDistribution ()**

Destructor.

Definition at line 26 of file DemandDistribution.cpp.

**10.14.3 Member Function Documentation****10.14.3.1 void TRADEMGEN::DemandDistribution::fromStream (std::istream & *ioIn*)**

Read a Business Object from an input stream.

**Parameters:**

*istream&* the input stream.

Definition at line 37 of file DemandDistribution.cpp.

**10.14.3.2 const std::string TRADEMGEN::DemandDistribution::describe () const**

Display of the structure.

Definition at line 41 of file DemandDistribution.cpp.

References `_meanNumberOfRequests`, and `_stdDevNumberOfRequests`.

Referenced by `TRADEMGENT::DemandStream::display()`, and `display()`.

**10.14.3.3 std::string TRADEMGEN::DemandDistribution::display () const**

Display demand distribution.

Definition at line 49 of file DemandDistribution.cpp.

References `describe()`.

**10.14.4 Member Data Documentation****10.14.4.1 std::NbOfRequests\_T [TRADEMGENT::DemandDistribution::\\_meanNumberOfRequests](#)**

Mean number of requests.

Definition at line 67 of file DemandDistribution.hpp.

Referenced by `describe()`, `TRADEMGENT::DemandStream::generateTimeOfRequestPoissonProcess()`, and `TRADEMGENT::DemandStream::getMeanNumberOfRequests()`.

**10.14.4.2 std::StdDevValue\_T [TRADEMGENT::DemandDistribution::\\_stdDevNumberOfRequests](#)**

Standard deviation of number of requests.

Definition at line 72 of file DemandDistribution.hpp.

Referenced by describe(), and TRADEMGEN::DemandStream::getStdDevNumberOfRequests().

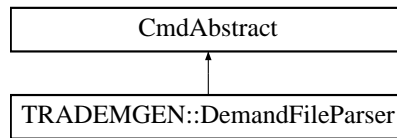
The documentation for this struct was generated from the following files:

- trademgen/basic/[DemandDistribution.hpp](#)
- trademgen/basic/[DemandDistribution.cpp](#)

## 10.15 TRADEMGEN::DemandFileParser Class Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandFileParser::



### Public Member Functions

- [DemandFileParser](#) (SEVMGR::SEVMGR\_ServicePtr\_T, stdair::RandomGeneration &, const [POSProbabilityMass\\_T](#) &, const stdair::Filename\_T & iDemandInputFilename)
- bool [generateDemand](#) ()

### 10.15.1 Detailed Description

Class wrapping the initialisation and entry point of the parser.

The seemingly redundancy is used to force the instantiation of the actual parser, which is a templatised Boost Spirit grammar. Hence, the actual parser is instantiated within that class object code.

Definition at line 434 of file DemandParserHelper.hpp.

### 10.15.2 Constructor & Destructor Documentation

**10.15.2.1 TRADEMGEN::DemandFileParser::DemandFileParser** (SEVMGR::SEVMGR\_ServicePtr\_T, stdair::RandomGeneration &, const [POSProbabilityMass\\_T](#) &, const stdair::Filename\_T & iDemandInputFilename)

Constructor.

### 10.15.3 Member Function Documentation

**10.15.3.1 bool TRADEMGEN::DemandFileParser::generateDemand ()**

Parse the demand input file.

Definition at line 897 of file DemandParserHelper.cpp.

Referenced by TRADEMGEN::DemandParser::generateDemand().

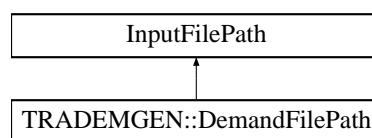
The documentation for this class was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.16 TRADEMGEN::DemandFilePath Class Reference

```
#include <trademgen/TRADEMGEN_Types.hpp>
```

Inheritance diagram for TRADEMGEN::DemandFilePath::



### Public Member Functions

- [DemandFilePath](#) (const stdair::Filename\_T &iFilename)

#### 10.16.1 Detailed Description

Demand input file.

Definition at line 30 of file TRADEMGEN\_Types.hpp.

#### 10.16.2 Constructor & Destructor Documentation

##### 10.16.2.1 TRADEMGEN::DemandFilePath::DemandFilePath (const stdair::Filename\_T & i-Filename) [inline, explicit]

Constructor.

Definition at line 35 of file TRADEMGEN\_Types.hpp.

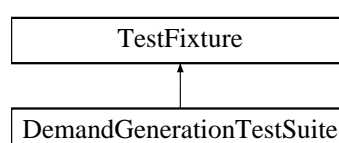
The documentation for this class was generated from the following file:

- trademgen/[TRADEMGEN\\_Types.hpp](#)

## 10.17 DemandGenerationTestSuite Class Reference

```
#include <test/trademgen/DemandGenerationTestSuite.hpp>
```

Inheritance diagram for DemandGenerationTestSuite::



**Public Member Functions**

- void [simpleEventGeneration](#) ()
- [DemandGenerationTestSuite](#) ()

**Protected Attributes**

- std::stringstream [\\_describeKey](#)

**10.17.1 Detailed Description**

Definition at line 6 of file DemandGenerationTestSuite.hpp.

**10.17.2 Constructor & Destructor Documentation****10.17.2.1 DemandGenerationTestSuite::DemandGenerationTestSuite ()**

Constructor.

**10.17.3 Member Function Documentation****10.17.3.1 void DemandGenerationTestSuite::simpleEventGeneration ()**

Test a simple event generation functionality.

**10.17.4 Member Data Documentation****10.17.4.1 std::stringstream DemandGenerationTestSuite::\_describeKey** [protected]

Definition at line 27 of file DemandGenerationTestSuite.hpp.

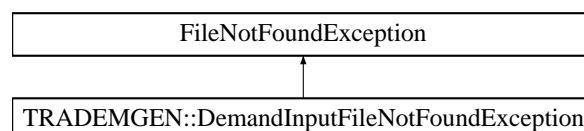
The documentation for this class was generated from the following file:

- test/trademgen/[DemandGenerationTestSuite.hpp](#)

**10.18 TRADEMGEN::DemandInputFileNotFoundException Class Reference**

```
#include <trademgen/TRADEMGEN_Exceptions.hpp>
```

Inheritance diagram for TRADEMGEN::DemandInputFileNotFoundException::

**Public Member Functions**

- [DemandInputFileNotFoundException](#) (const std::string &iWhat)



### 10.18.1 Detailed Description

Exception when no demand input file can be found

Definition at line 30 of file TRADEMGEN\_Exceptions.hpp.

### 10.18.2 Constructor & Destructor Documentation

#### 10.18.2.1 TRADEMGEN::DemandInputFileNotFoundException::DemandInputFileNotFoundException (const std::string & *iWhat*) [inline]

Constructor.

Definition at line 36 of file TRADEMGEN\_Exceptions.hpp.

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

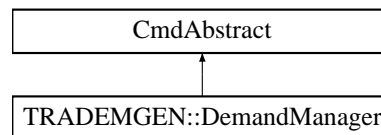
- [trademgen/TRADEMGEN\\_Exceptions.hpp](#)

## 10.19 TRADEMGEN::DemandManager Class Reference

Utility class for Demand and [DemandStream](#) objects.

```
#include <trademgen/command/DemandManager.hpp>
```

Inheritance diagram for TRADEMGEN::DemandManager::



### Friends

- struct [DemandParserHelper::doEndDemand](#)
- class [TRADEMGEN\\_Service](#)

### 10.19.1 Detailed Description

Utility class for Demand and [DemandStream](#) objects.

Definition at line 40 of file DemandManager.hpp.

### 10.19.2 Friends And Related Function Documentation

#### 10.19.2.1 friend struct [DemandParserHelper::doEndDemand](#) [friend]

Definition at line 41 of file DemandManager.hpp.

#### 10.19.2.2 friend class [TRADEMGEN\\_Service](#) [friend]

Definition at line 42 of file DemandManager.hpp.

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

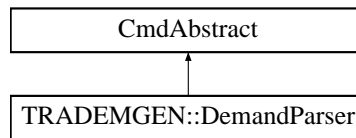
- trademgen/command/[DemandManager.hpp](#)
- trademgen/command/[DemandManager.cpp](#)

## 10.20 TRADEMGEN::DemandParser Class Reference

Class wrapping the parser entry point.

```
#include <trademgen/command/DemandParser.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParser:



### Static Public Member Functions

- static void [generateDemand](#) (const [DemandFilePath](#) &, SEVMGR::SEVMGR\_ServicePtr\_T, stdair::RandomGeneration &, const [POSProbabilityMass\\_T](#) &)

#### 10.20.1 Detailed Description

Class wrapping the parser entry point.

Definition at line 28 of file DemandParser.hpp.

#### 10.20.2 Member Function Documentation

**10.20.2.1 void TRADEMGEN::DemandParser::generateDemand (const [DemandFilePath](#) &, SEVMGR::SEVMGR\_ServicePtr\_T, stdair::RandomGeneration &, const [POSProbabilityMass\\_T](#) &) [static]**

Parse the CSV file describing travel demand, for instance for generating simulated booking request in a simulator.

The state of the random generator, given as parameter, evolves each time a demand request is generated.

#### Parameters:

**const [DemandFilePath](#)&** The file-name of the CSV-formatted demand input file.

**SEVMGR::SEVMGR\_ServicePtr\_T** Pointer on the SEvMgr service handler to update the queue with the parsed information.

**stdair::RandomGeneration&** Random generator.

Definition at line 18 of file DemandParser.cpp.

References TRADEMGEN::DemandFileParser::generateDemand().

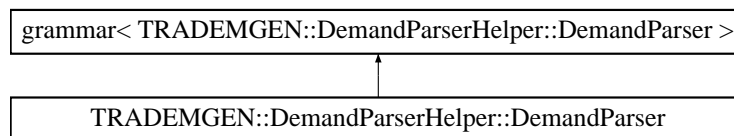
The documentation for this class was generated from the following files:

- [trademgen/command/DemandParser.hpp](#)
- [trademgen/command/DemandParser.cpp](#)

## 10.21 TRADEMGEN::DemandParserHelper::DemandParser Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::DemandParser::



### Public Member Functions

- [DemandParser](#) (SEVMGR::SEVMGR\_ServicePtr\_T, stdair::RandomGeneration &, const [POSProbabilityMass\\_T](#) &, [DemandStruct](#) &)

### Public Attributes

- SEVMGR::SEVMGR\_ServicePtr\_T [\\_sevmgrServicePtr](#)
- stdair::RandomGeneration & [\\_uniformGenerator](#)
- const [POSProbabilityMass\\_T](#) & [\\_posProbabilityMass](#)
- [DemandStruct](#) & [\\_demand](#)

### Classes

- struct [definition](#)

#### 10.21.1 Detailed Description

Grammar for the demand parser.

Definition at line 384 of file DemandParserHelper.hpp.

#### 10.21.2 Constructor & Destructor Documentation

**10.21.2.1 TRADEMGEN::DemandParserHelper::DemandParser::DemandParser**  
(SEVMGR::SEVMGR\_ServicePtr\_T, stdair::RandomGeneration &, const [POSProbabilityMass\\_T](#) &, [DemandStruct](#) &)

Definition at line 568 of file DemandParserHelper.cpp.

#### 10.21.3 Member Data Documentation

**10.21.3.1 SEVMGR::SEVMGR\_ServicePtr\_T TRADEMGEN::DemandParserHelper::DemandParser::\_sevmgrServicePtr**

## 10.22 TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT > Struct Template Reference 59

---

Definition at line 415 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

### 10.21.3.2 stdair::RandomGeneration& [TRADEMGEN::DemandParserHelper::DemandParser::\\_uniformGenerator](#)

Definition at line 416 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

### 10.21.3.3 const [POSProbabilityMass\\_T](#)& [TRADEMGEN::DemandParserHelper::DemandParser::\\_posProbabilityMass](#)

Definition at line 417 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

### 10.21.3.4 [DemandStruct](#)& [TRADEMGEN::DemandParserHelper::DemandParser::\\_demand](#)

Definition at line 418 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

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

- [trademgen/command/DemandParserHelper.hpp](#)
- [trademgen/command/DemandParserHelper.cpp](#)

## 10.22 TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT > Struct Template Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

### Public Member Functions

- [definition](#) ([DemandParser](#) const &self)
- [boost::spirit::classic::rule< ScannerT > const & start](#) () const

### Public Attributes

- [boost::spirit::classic::rule< ScannerT > demand\\_list](#)
- [boost::spirit::classic::rule< ScannerT > not\\_to\\_be\\_parsed](#)
- [boost::spirit::classic::rule< ScannerT > demand](#)
- [boost::spirit::classic::rule< ScannerT > demand\\_end](#)
- [boost::spirit::classic::rule< ScannerT > pref\\_dep\\_date\\_range](#)
- [boost::spirit::classic::rule< ScannerT > date](#)
- [boost::spirit::classic::rule< ScannerT > dow](#)
- [boost::spirit::classic::rule< ScannerT > origin](#)

- boost::spirit::classic::rule< ScannerT > [destination](#)
- boost::spirit::classic::rule< ScannerT > [pref\\_cabin](#)
- boost::spirit::classic::rule< ScannerT > [demand\\_params](#)
- boost::spirit::classic::rule< ScannerT > [pos\\_dist](#)
- boost::spirit::classic::rule< ScannerT > [pos\\_pair](#)
- boost::spirit::classic::rule< ScannerT > [pos\\_code](#)
- boost::spirit::classic::rule< ScannerT > [pos\\_share](#)
- boost::spirit::classic::rule< ScannerT > [channel\\_dist](#)
- boost::spirit::classic::rule< ScannerT > [channel\\_pair](#)
- boost::spirit::classic::rule< ScannerT > [channel\\_code](#)
- boost::spirit::classic::rule< ScannerT > [channel\\_share](#)
- boost::spirit::classic::rule< ScannerT > [trip\\_dist](#)
- boost::spirit::classic::rule< ScannerT > [trip\\_pair](#)
- boost::spirit::classic::rule< ScannerT > [trip\\_code](#)
- boost::spirit::classic::rule< ScannerT > [trip\\_share](#)
- boost::spirit::classic::rule< ScannerT > [stay\\_dist](#)
- boost::spirit::classic::rule< ScannerT > [stay\\_pair](#)
- boost::spirit::classic::rule< ScannerT > [stay\\_share](#)
- boost::spirit::classic::rule< ScannerT > [ff\\_dist](#)
- boost::spirit::classic::rule< ScannerT > [ff\\_pair](#)
- boost::spirit::classic::rule< ScannerT > [ff\\_code](#)
- boost::spirit::classic::rule< ScannerT > [ff\\_share](#)
- boost::spirit::classic::rule< ScannerT > [change\\_fees](#)
- boost::spirit::classic::rule< ScannerT > [non\\_refundable](#)
- boost::spirit::classic::rule< ScannerT > [pref\\_dep\\_time\\_dist](#)
- boost::spirit::classic::rule< ScannerT > [pref\\_dep\\_time\\_pair](#)
- boost::spirit::classic::rule< ScannerT > [pref\\_dep\\_time\\_share](#)
- boost::spirit::classic::rule< ScannerT > [time](#)
- boost::spirit::classic::rule< ScannerT > [wtp](#)
- boost::spirit::classic::rule< ScannerT > [time\\_value\\_dist](#)
- boost::spirit::classic::rule< ScannerT > [time\\_value\\_pair](#)
- boost::spirit::classic::rule< ScannerT > [time\\_value\\_share](#)
- boost::spirit::classic::rule< ScannerT > [dtd\\_dist](#)
- boost::spirit::classic::rule< ScannerT > [dtd\\_pair](#)
- boost::spirit::classic::rule< ScannerT > [dtd\\_share](#)

### 10.22.1 Detailed Description

**template<typename ScannerT> struct TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >**

Definition at line 391 of file DemandParserHelper.hpp.

## 10.22.2 Constructor & Destructor Documentation

### 10.22.2.1 template<typename ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition (DemandParser const & self)

Definition at line 580 of file DemandParserHelper.cpp.

References TRADEMGEN::DemandParserHelper::DemandParser::\_demand, TRADEMGEN::DemandParserHelper::DemandParser::\_posProbabilityMass, TRADEMGEN::DemandParserHelper::DemandParser::\_sevmgrServicePtr, TRADEMGEN::DemandParserHelper::DemandParser::\_uniformGenerator, TRADEMGEN::DemandParserHelper::airport\_p(), TRADEMGEN::DemandParserHelper::cabin\_code\_p(), TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::change\_fees, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::channel\_code, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::channel\_dist, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::channel\_pair, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::channel\_share, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::date, TRADEMGEN::DemandParserHelper::day\_p(), TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::demand, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::demand\_end, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::demand\_list, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::demand\_params, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::destination, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::dow, TRADEMGEN::DemandParserHelper::dow\_p(), TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::dtd\_dist, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::dtd\_pair, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::dtd\_share, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::ff\_code, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::ff\_dist, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::ff\_pair, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::ff\_share, TRADEMGEN::DemandParserHelper::ff\_type\_p(), TRADEMGEN::DemandParserHelper::hours\_p(), TRADEMGEN::DemandParserHelper::minutes\_p(), TRADEMGEN::DemandParserHelper::month\_p(), TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::non\_refundable, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::not\_to\_be\_parsed, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::origin, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::pos\_code, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::pos\_dist, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::pos\_pair, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::pos\_share, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::pref\_cabin, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::pref\_dep\_date\_range, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::pref\_dep\_time\_dist, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::pref\_dep\_time\_pair, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::pref\_dep\_time\_share, TRADEMGEN::DemandParserHelper::seconds\_p(), TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::stay\_dist, TRADEMGEN::DemandParserHelper::stay\_duration\_p(), TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::stay\_pair, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::stay\_share, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::time, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::time\_value\_dist, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::time\_value\_pair, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::time\_value\_share, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::trip\_code, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::trip\_

dist, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::trip\_  
pair, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::trip\_  
share, TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::wtp, and  
TRADEMGENT::DemandParserHelper::year\_p().

### 10.22.3 Member Function Documentation

#### 10.22.3.1 `template<typename ScannerT> bsc::rule< ScannerT > const & TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::start () const`

Entry point of the parser.

Definition at line 839 of file DemandParserHelper.cpp.

References TRADEMGENT::DemandParserHelper::DemandParser::definition< ScannerT >::demand\_list.

### 10.22.4 Member Data Documentation

#### 10.22.4.1 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::demand_list`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGENT::DemandParserHelper::DemandParser::definition< ScannerT >::definition(), and TRADEMGENT::DemandParserHelper::DemandParser::definition< ScannerT >::start().

#### 10.22.4.2 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::not_to_be_parsed`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGENT::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

#### 10.22.4.3 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::demand`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGENT::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

#### 10.22.4.4 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::demand_end`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGENT::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

#### 10.22.4.5 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::pref_dep_date_range`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.6 template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::date**

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.7 template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::dow**

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.8 template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::origin**

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.9 template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::destination**

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.10 template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::pref\_cabin**

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.11 template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::demand\_params**

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().



**10.22.4.12** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::pos_dist`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.13** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::pos_pair`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.14** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::pos_code`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.15** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::pos_share`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.16** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::channel_dist`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.17** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::channel_pair`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.18** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::channel_code`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.19** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::channel_share`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.20** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::trip_dist`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.21** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::trip_pair`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.22** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::trip_code`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.23** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::trip_share`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.24** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::stay_dist`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.25** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::stay_pair`

Definition at line 395 of file DemandParserHelper.hpp.

## 10.22 TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT > Struct Template Reference

66

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.26** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::stay_share`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.27** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::ff_dist`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.28** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::ff_pair`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.29** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::ff_code`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.30** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::ff_share`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.31** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::change_fees`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.32** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::non_refundable`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.33** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::pref_dep_time_dist`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.34** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::pref_dep_time_pair`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.35** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::pref_dep_time_share`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.36** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::time`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.37** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::wtp`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.38** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::time_value_dist`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.39** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::time_value_pair`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.40** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::time_value_share`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.41** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::dtd_dist`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.42** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::dtd_pair`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

**10.22.4.43** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::dtd_share`

Definition at line 395 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >::definition().

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

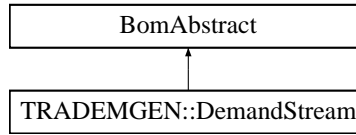
- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.23 TRADEMGEN::DemandStream Class Reference

Class modeling a demand stream.

```
#include <trademgen/bom/DemandStream.hpp>
```

Inheritance diagram for TRADEMGEN::DemandStream::



## Public Types

- typedef [DemandStreamKey](#) [Key\\_T](#)

## Public Member Functions

- const [Key\\_T](#) & [getKey](#) () const
- [BomAbstract](#) \*const [getParent](#) () const
- const stdair::AirportCode\_T & [getOrigin](#) () const
- const stdair::AirportCode\_T & [getDestination](#) () const
- const stdair::Date\_T & [getPreferredDepartureDate](#) () const
- const stdair::CabinCode\_T & [getPreferredCabin](#) () const
- const stdair::HolderMap\_T & [getHolderMap](#) () const
- const [DemandCharacteristics](#) & [getDemandCharacteristics](#) () const
- const [DemandDistribution](#) & [getDemandDistribution](#) () const
- const stdair::NbOfRequests\_T & [getTotalNumberOfRequestsToBeGenerated](#) () const
- const stdair::NbOfRequests\_T & [getMeanNumberOfRequests](#) () const
- const stdair::StdDevValue\_T & [getStdDevNumberOfRequests](#) () const
- const stdair::Count\_T & [getNumberOfRequestsGeneratedSoFar](#) () const
- const stdair::Disutility\_T & [getChangeFeeDisutility](#) () const
- const stdair::Disutility\_T & [getNonRefundableDisutility](#) () const
- const [POSProbabilityMass\\_T](#) & [getPOSProbabilityMass](#) () const
- void [setNumberOfRequestsGeneratedSoFar](#) (const stdair::Count\_T &iCount)
- void [setDemandDistribution](#) (const [DemandDistribution](#) &iDemandDistribution)
- void [setDemandCharacteristics](#) (const [ArrivalPatternCumulativeDistribution\\_T](#) &iArrivalPattern, const [POSProbabilityMassFunction\\_T](#) &iPOSProbMass, const [ChannelProbabilityMassFunction\\_T](#) &iChannelProbMass, const [TripTypeProbabilityMassFunction\\_T](#) &iTripTypeProbMass, const [StayDurationProbabilityMassFunction\\_T](#) &iStayDurationProbMass, const [FrequentFlyerProbabilityMassFunction\\_T](#) &iFrequentFlyerProbMass, const stdair::ChangeFeesRatio\_T &iChangeFeeProb, const stdair::Disutility\_T &iChangeFeeDisutility, const stdair::NonRefundableRatio\_T &iNonRefundableProb, const stdair::Disutility\_T &iNonRefundableDisutility, const [PreferredDepartureTimeContinuousDistribution\\_T](#) &iPreferredDepartureTimeContinuousDistribution, const stdair::WTP\_T &iMinWTP, const [ValueOfTimeContinuousDistribution\\_T](#) &iValueOfTimeContinuousDistribution)
- void [setTotalNumberOfRequestsToBeGenerated](#) (const stdair::NbOfRequests\_T &iNbOfRequests)
- void [setRequestDateTimeRandomGeneratorSeed](#) (const stdair::RandomSeed\_T &iSeed)
- void [setDemandCharacteristicsRandomGeneratorSeed](#) (const stdair::RandomSeed\_T &iSeed)
- void [setPOSProbabilityMass](#) (const [POSProbabilityMass\\_T](#) &iProbMass)
- void [setAll](#) (const [ArrivalPatternCumulativeDistribution\\_T](#) &, const [POSProbabilityMassFunction\\_T](#) &, const [ChannelProbabilityMassFunction\\_T](#) &, const [TripTypeProbabilityMassFunction\\_T](#) &, const [StayDurationProbabilityMassFunction\\_T](#) &, const [FrequentFlyerProbabilityMassFunction\\_T](#) &)

- &, const stdair::ChangeFeesRatio\_T &, const stdair::Disutility\_T &, const stdair::NonRefundableRatio\_T &, const stdair::Disutility\_T &, const PreferredDepartureTimeContinuousDistribution\_T &, const stdair::WTP\_T &, const ValueOfTimeContinuousDistribution\_T &, const DemandDistribution &, stdair::BaseGenerator\_T &ioSharedGenerator, const stdair::RandomSeed\_T &iRequestDate-TimeSeed, const stdair::RandomSeed\_T &iDemandCharacteristicsSeed, const POSProbabilityMass\_T &)
- void [setBoolFirstDateTimeRequest](#) (const bool &iFirstDateTimeRequest)
- void [incrementGeneratedRequestsCounter](#) ()
- const bool [stillHavingRequestsToBeGenerated](#) (const stdair::DemandGenerationMethod &iDemandGenerationMethod) const
- const stdair::DateTime\_T [generateTimeOfRequestPoissonProcess](#) ()
- const stdair::DateTime\_T [generateTimeOfRequestStatisticsOrder](#) ()
- const stdair::AirportCode\_T [generatePOS](#) ()
- const stdair::ChannelLabel\_T [generateChannel](#) ()
- const stdair::TripType\_T [generateTripType](#) ()
- const stdair::DayDuration\_T [generateStayDuration](#) ()
- const stdair::FrequentFlyer\_T [generateFrequentFlyer](#) ()
- const stdair::ChangeFees\_T [generateChangeFees](#) ()
- const stdair::NonRefundable\_T [generateNonRefundable](#) ()
- const stdair::Duration\_T [generatePreferredDepartureTime](#) ()
- const stdair::WTP\_T [generateWTP](#) (stdair::RandomGeneration &, const stdair::Date\_T &, const stdair::DateTime\_T &, const stdair::DayDuration\_T &)
- const stdair::PriceValue\_T [generateValueOfTime](#) ()
- stdair::BookingRequestPtr\_T [generateNextRequest](#) (stdair::RandomGeneration &, const stdair::DemandGenerationMethod &)
- void [reset](#) (stdair::BaseGenerator\_T &ioSharedGenerator)
- void [toStream](#) (std::ostream &ioOut) const
- void [fromStream](#) (std::istream &ioIn)
- std::string [toString](#) () const
- const std::string [describeKey](#) () const
- std::string [display](#) () const
- const stdair::Duration\_T [convertFloatIntoDuration](#) (const stdair::FloatDuration\_T)

### Protected Member Functions

- [DemandStream](#) (const Key\_T &)
- virtual [~DemandStream](#) ()

### Protected Attributes

- [Key\\_T\\_key](#)
- [BomAbstract](#) \* [\\_parent](#)
- stdair::HolderMap\_T [\\_holderMap](#)
- [DemandCharacteristics](#) [\\_demandCharacteristics](#)
- [DemandDistribution](#) [\\_demandDistribution](#)
- stdair::NbOfRequests\_T [\\_totalNumberOfRequestsToBeGenerated](#)
- [RandomGenerationContext](#) [\\_randomGenerationContext](#)
- stdair::RandomGeneration [\\_requestDateTimeRandomGenerator](#)
- stdair::RandomGeneration [\\_demandCharacteristicsRandomGenerator](#)
- [POSProbabilityMass\\_T](#) [\\_posProMass](#)

## Friends

- class [stdair::FacBom](#)
- class [stdair::FacBomManager](#)

### 10.23.1 Detailed Description

Class modeling a demand stream.

Definition at line 30 of file DemandStream.hpp.

### 10.23.2 Member Typedef Documentation

#### 10.23.2.1 typedef [DemandStreamKey](#) TRADEMGEN::DemandStream::Key\_T

Definition allowing to retrieve the associated BOM key type.

Definition at line 39 of file DemandStream.hpp.

### 10.23.3 Constructor & Destructor Documentation

#### 10.23.3.1 TRADEMGEN::DemandStream::DemandStream (const [Key\\_T](#) &) [protected]

Main constructor.

Definition at line 64 of file DemandStream.cpp.

#### 10.23.3.2 TRADEMGEN::DemandStream::~~DemandStream () [protected, virtual]

Destructor.

Definition at line 69 of file DemandStream.cpp.

### 10.23.4 Member Function Documentation

#### 10.23.4.1 const [Key\\_T](#)& TRADEMGEN::DemandStream::getKey () const [inline]

Get the key

Definition at line 45 of file DemandStream.hpp.

References `_key`.

#### 10.23.4.2 [BomAbstract\\*](#) const TRADEMGEN::DemandStream::getParent () const [inline]

Get the parent object (EventQueue).

Definition at line 50 of file DemandStream.hpp.

References `_parent`.

#### 10.23.4.3 const [stdair::AirportCode\\_T](#)& TRADEMGEN::DemandStream::getOrigin () const [inline]

Get the origin (part of the primary key).



Definition at line 55 of file DemandStream.hpp.

References `_key`, and `TRADEMGEN::DemandStreamKey::getOrigin()`.

#### 10.23.4.4 `const stdair::AirportCode_T& TRADEMGEN::DemandStream::getDestination () const [inline]`

Get the destination (part of the primary key).

Definition at line 60 of file DemandStream.hpp.

References `_key`, and `TRADEMGEN::DemandStreamKey::getDestination()`.

#### 10.23.4.5 `const stdair::Date_T& TRADEMGEN::DemandStream::getPreferredDepartureDate () const [inline]`

Get the preferred departure date (part of the primary key).

Definition at line 65 of file DemandStream.hpp.

References `_key`, and `TRADEMGEN::DemandStreamKey::getPreferredDepartureDate()`.

#### 10.23.4.6 `const stdair::CabinCode_T& TRADEMGEN::DemandStream::getPreferredCabin () const [inline]`

Get the preferred cabin (part of the primary key).

Definition at line 70 of file DemandStream.hpp.

References `_key`, and `TRADEMGEN::DemandStreamKey::getPreferredCabin()`.

#### 10.23.4.7 `const stdair::HolderMap_T& TRADEMGEN::DemandStream::getHolderMap () const [inline]`

Get the map of children holders.

Definition at line 75 of file DemandStream.hpp.

References `_holderMap`.

#### 10.23.4.8 `const DemandCharacteristics& TRADEMGEN::DemandStream::getDemandCharacteristics () const [inline]`

Get the demand characteristics.

Definition at line 80 of file DemandStream.hpp.

References `_demandCharacteristics`.

#### 10.23.4.9 `const DemandDistribution& TRADEMGEN::DemandStream::getDemandDistribution () const [inline]`

Get the demand distribution.

Definition at line 85 of file DemandStream.hpp.

References `_demandDistribution`.

**10.23.4.10 const stdair::NbOfRequests\_T& TRADEMGEN::DemandStream::getTotalNumberOfRequestsToBeGenerated () const [inline]**

Get the total number of requests to be generated.

Definition at line 90 of file DemandStream.hpp.

References `_totalNumberOfRequestsToBeGenerated`.

**10.23.4.11 const stdair::NbOfRequests\_T& TRADEMGEN::DemandStream::getMeanNumberOfRequests () const [inline]**

Get the mean (expected) number of requests.

Definition at line 95 of file DemandStream.hpp.

References `_demandDistribution`, and `TRADEMGENT::DemandDistribution::_meanNumberOfRequests`.

**10.23.4.12 const stdair::StdDevValue\_T& TRADEMGEN::DemandStream::getStdDevNumberOfRequests () const [inline]**

Get the standard deviation of number of requests.

Definition at line 100 of file DemandStream.hpp.

References `_demandDistribution`, and `TRADEMGENT::DemandDistribution::_stdDevNumberOfRequests`.

**10.23.4.13 const stdair::Count\_T& TRADEMGEN::DemandStream::getNumberOfRequestsGeneratedSoFar () const [inline]**

Get the number of requests generated so far.

Definition at line 105 of file DemandStream.hpp.

References `_randomGenerationContext`, and `TRADEMGENT::RandomGenerationContext::getNumberOfRequestsGeneratedSoFar()`.

**10.23.4.14 const stdair::Disutility\_T& TRADEMGEN::DemandStream::getChangeFeeDisutility () const [inline]**

Get the change fee disutility.

Definition at line 110 of file DemandStream.hpp.

References `TRADEMGENT::DemandCharacteristics::_changeFeeDisutility`, and `_demandCharacteristics`.

**10.23.4.15 const stdair::Disutility\_T& TRADEMGEN::DemandStream::getNonRefundableDisutility () const [inline]**

Get the non refundable disutility.

Definition at line 115 of file DemandStream.hpp.

References `_demandCharacteristics`, and `TRADEMGENT::DemandCharacteristics::_nonRefundableDisutility`.

**10.23.4.16 const POSProbabilityMass\_T& TRADEMGEN::DemandStream::getPOSProbabilityMass () const [inline]**

Get the default POS probability mass, used when "row" (rest of the world) is drawn.

Definition at line 123 of file DemandStream.hpp.

References `_posProMass`.

**10.23.4.17 void TRADEMGEN::DemandStream::setNumberOfRequestsGeneratedSoFar (const stdair::Count\_T & iCount) [inline]**

Set the number of requests generated so far.

Definition at line 131 of file DemandStream.hpp.

References `_randomGenerationContext`, and `TRADEMGENT::RandomGenerationContext::setNumberOfRequestsGeneratedSoFar()`.

**10.23.4.18 void TRADEMGEN::DemandStream::setDemandDistribution (const Demand-Distribution & iDemandDistribution) [inline]**

Set the demand distribution.

Definition at line 136 of file DemandStream.hpp.

References `_demandDistribution`.

Referenced by `setAll()`.

**10.23.4.19 void TRADEMGEN::DemandStream::setDemandCharacteristics (const Arrival-PatternCumulativeDistribution\_T & iArrivalPattern, const POSProbabilityMassFunction\_T & iPOSProbMass, const ChannelProbabilityMassFunction\_T & iChannelProbMass, const TripType-ProbabilityMassFunction\_T & iTripTypeProbMass, const StayDurationProbabilityMassFunction\_T & iStayDurationProbMass, const FrequentFlyerProbabilityMassFunction\_T & iFrequentFlyer-ProbMass, const stdair::ChangeFeesRatio\_T & iChangeFeeProb, const stdair::Disutility\_T & iChangeFeeDisutility, const stdair::NonRefundableRatio\_T & iNonRefundableProb, const stdair::Disutility\_T & iNonRefundableDisutility, const PreferredDepartureTimeContinuous-Distribution\_T & iPreferredDepartureTimeContinuousDistribution, const stdair::WTP\_T & iMinWTP, const ValueOfTimeContinuousDistribution\_T & iValueOfTimeContinuousDistribution) [inline]**

Set the demand characteristics.

Definition at line 142 of file DemandStream.hpp.

References `_demandCharacteristics`.

Referenced by `setAll()`.

**10.23.4.20 void TRADEMGEN::DemandStream::setTotalNumberOfRequestsToBeGenerated (const stdair::NbOfRequests\_T & iNbOfRequests) [inline]**

Set the total number of requests to be generated.

Definition at line 166 of file DemandStream.hpp.

References `_totalNumberOfRequestsToBeGenerated`.

Referenced by `setAll()`.

**10.23.4.21 void TRADEMGEN::DemandStream::setRequestDateTimeRandomGeneratorSeed (const stdair::RandomSeed\_T & iSeed) [inline]**

Set the seed of the random generator for the request datetime.

Definition at line 171 of file DemandStream.hpp.

References `_requestDateTimeRandomGenerator`.

Referenced by `setAll()`.

**10.23.4.22 void TRADEMGEN::DemandStream::setDemandCharacteristicsRandomGeneratorSeed (const stdair::RandomSeed\_T & iSeed) [inline]**

Set the seed of the random generator for the demand characteristics.

Definition at line 176 of file DemandStream.hpp.

References `_demandCharacteristicsRandomGenerator`.

Referenced by `setAll()`.

**10.23.4.23 void TRADEMGEN::DemandStream::setPOSProbabilityMass (const POSProbabilityMass\_T & iProbMass) [inline]**

Set the default POS probability mass, used when "row" (rest of the world) is drawn.

Definition at line 184 of file DemandStream.hpp.

References `_posProMass`.

Referenced by `setAll()`.

**10.23.4.24 void TRADEMGEN::DemandStream::setAll (const ArrivalPatternCumulativeDistribution\_T &, const POSProbabilityMassFunction\_T &, const ChannelProbabilityMassFunction\_T &, const TripTypeProbabilityMassFunction\_T &, const StayDurationProbabilityMassFunction\_T &, const FrequentFlyerProbabilityMassFunction\_T &, const stdair::ChangeFeesRatio\_T &, const stdair::Disutility\_T &, const stdair::NonRefundableRatio\_T &, const stdair::Disutility\_T &, const PreferredDepartureTimeContinuousDistribution\_T &, const stdair::WTP\_T &, const ValueOfTimeContinuousDistribution\_T &, const DemandDistribution &, stdair::BaseGenerator\_T & ioSharedGenerator, const stdair::RandomSeed\_T & iRequestDateTimeSeed, const stdair::RandomSeed\_T & iDemandCharacteristicsSeed, const POSProbabilityMass\_T &)**

Initialisation.

Definition at line 81 of file DemandStream.cpp.

References `setDemandCharacteristics()`, `setDemandCharacteristicsRandomGeneratorSeed()`, `setDemandDistribution()`, `setPOSProbabilityMass()`, `setRequestDateTimeRandomGeneratorSeed()`, and `setTotalNumberOfRequestsToBeGenerated()`.

**10.23.4.25 void TRADEMGEN::DemandStream::setBoolFirstDateTimeRequest (const bool & iFirstDateTimeRequest) [inline]**

Set the boolean describing if it is the first time we generate a request for a demand stream.

Definition at line 214 of file DemandStream.hpp.

#### 10.23.4.26 void TRADEMGEN::DemandStream::incrementGeneratedRequestsCounter () [inline]

Increment counter of requests generated so far

Definition at line 222 of file DemandStream.hpp.

References `_randomGenerationContext`, and `TRADEMGEN::RandomGenerationContext::incrementGeneratedRequestsCounter()`.

Referenced by `generateTimeOfRequestPoissonProcess()`, and `generateTimeOfRequestStatisticsOrder()`.

#### 10.23.4.27 const bool TRADEMGEN::DemandStream::stillHavingRequestsToBeGenerated (const stdair::DemandGenerationMethod & iDemandGenerationMethod) const

Check whether enough requests have already been generated.

Definition at line 172 of file DemandStream.cpp.

References `_randomGenerationContext`, `_totalNumberOfRequestsToBeGenerated`, and `TRADEMGEN::RandomGenerationContext::getNumberOfRequestsGeneratedSoFar()`.

#### 10.23.4.28 const stdair::DateTime\_T TRADEMGEN::DemandStream::generateTimeOfRequestPoissonProcess ()

Generate the time of the next request with poisson process.

Definition at line 197 of file DemandStream.cpp.

References `TRADEMGEN::DemandCharacteristics::_arrivalPattern`, `_demandCharacteristics`, `_demandDistribution`, `_key`, `TRADEMGEN::DemandDistribution::_meanNumberOfRequests`, `_requestDateTimeRandomGenerator`, `convertFloatIntoDuration()`, `TRADEMGEN::DEFAULT_LAST_LOWER_BOUND_ARRIVAL_PATTERN`, `TRADEMGEN::ContinuousAttributeLite< T >::getDerivativeValue()`, `TRADEMGEN::DemandStreamKey::getPreferredDepartureDate()`, `TRADEMGEN::ContinuousAttributeLite< T >::getUpperBound()`, `TRADEMGEN::ContinuousAttributeLite< T >::getValue()`, and `incrementGeneratedRequestsCounter()`.

Referenced by `generateNextRequest()`.

#### 10.23.4.29 const stdair::DateTime\_T TRADEMGEN::DemandStream::generateTimeOfRequestStatisticsOrder ()

Generate the time of the next request with statistics order

Definition at line 299 of file DemandStream.cpp.

References `TRADEMGEN::DemandCharacteristics::_arrivalPattern`, `_demandCharacteristics`, `_key`, `_randomGenerationContext`, `_requestDateTimeRandomGenerator`, `_totalNumberOfRequestsToBeGenerated`, `convertFloatIntoDuration()`, `TRADEMGEN::RandomGenerationContext::getCumulativeProbabilitySoFar()`, `TRADEMGEN::RandomGenerationContext::getNumberOfRequestsGeneratedSoFar()`, `TRADEMGEN::DemandStreamKey::getPreferredDepartureDate()`, `TRADEMGEN::ContinuousAttributeLite< T >::getValue()`, `incrementGeneratedRequestsCounter()`, and `TRADEMGEN::RandomGenerationContext::setCumulativeProbabilitySoFar()`.

Referenced by `generateNextRequest()`.

#### 10.23.4.30 const stdair::AirportCode\_T TRADEMGEN::DemandStream::generatePOS ()

Generate the POS.

Definition at line 430 of file DemandStream.cpp.

References `_demandCharacteristics`, `_demandCharacteristicsRandomGenerator`, and `TRADEMGEN::DemandCharacteristics::getPOSValue()`.

Referenced by `generateNextRequest()`.

#### 10.23.4.31 `const stdair::ChannelLabel_T TRADEMGEN::DemandStream::generateChannel ()`

Generate the reservation channel.

Definition at line 440 of file DemandStream.cpp.

References `TRADEMGEN::DemandCharacteristics::_channelProbabilityMass`, `_demandCharacteristics`, `_demandCharacteristicsRandomGenerator`, and `TRADEMGEN::CategoricalAttributeLite< T >::getValue()`.

Referenced by `generateNextRequest()`.

#### 10.23.4.32 `const stdair::TripType_T TRADEMGEN::DemandStream::generateTripType ()`

Generate the trip type.

Definition at line 449 of file DemandStream.cpp.

References `_demandCharacteristics`, `_demandCharacteristicsRandomGenerator`, `TRADEMGEN::DemandCharacteristics::_tripTypeProbabilityMass`, and `TRADEMGEN::CategoricalAttributeLite< T >::getValue()`.

Referenced by `generateNextRequest()`.

#### 10.23.4.33 `const stdair::DayDuration_T TRADEMGEN::DemandStream::generateStayDuration ()`

Generate the stay duration.

Definition at line 458 of file DemandStream.cpp.

References `_demandCharacteristics`, `_demandCharacteristicsRandomGenerator`, `TRADEMGEN::DemandCharacteristics::_stayDurationProbabilityMass`, and `TRADEMGEN::CategoricalAttributeLite< T >::getValue()`.

Referenced by `generateNextRequest()`.

#### 10.23.4.34 `const stdair::FrequentFlyer_T TRADEMGEN::DemandStream::generateFrequentFlyer ()`

Generate the frequent flyer type.

Definition at line 467 of file DemandStream.cpp.

References `_demandCharacteristics`, `_demandCharacteristicsRandomGenerator`, `TRADEMGEN::DemandCharacteristics::_frequentFlyerProbabilityMass`, and `TRADEMGEN::CategoricalAttributeLite< T >::getValue()`.

Referenced by `generateNextRequest()`.

#### 10.23.4.35 `const stdair::ChangeFees_T TRADEMGEN::DemandStream::generateChangeFees ()`

Generate the change fee acceptance.

Definition at line 476 of file DemandStream.cpp.

References TRADEMGEN::DemandCharacteristics::\_changeFeeProb, \_demandCharacteristics, and \_demandCharacteristicsRandomGenerator.

Referenced by generateNextRequest().

#### 10.23.4.36 const stdair::NonRefundable\_T TRADEMGEN::DemandStream::generateNonRefundable ()

Generate the non refundable acceptance.

Definition at line 487 of file DemandStream.cpp.

References \_demandCharacteristics, \_demandCharacteristicsRandomGenerator, and TRADEMGEN::DemandCharacteristics::\_nonRefundableProb.

Referenced by generateNextRequest().

#### 10.23.4.37 const stdair::Duration\_T TRADEMGEN::DemandStream::generatePreferredDepartureTime ()

Generate the preferred departure time.

Definition at line 498 of file DemandStream.cpp.

References \_demandCharacteristics, and \_demandCharacteristicsRandomGenerator.

Referenced by generateNextRequest().

#### 10.23.4.38 const stdair::WTP\_T TRADEMGEN::DemandStream::generateWTP (stdair::RandomGeneration &, const stdair::Date\_T &, const stdair::DateTime\_T &, const stdair::DayDuration\_T &)

Generate the WTP.

Definition at line 512 of file DemandStream.cpp.

References \_demandCharacteristics, TRADEMGEN::DemandCharacteristics::\_frat5Pattern, TRADEMGEN::DemandCharacteristics::\_minWTP, and TRADEMGEN::ContinuousAttributeLite< T >::getValue().

Referenced by generateNextRequest().

#### 10.23.4.39 const stdair::PriceValue\_T TRADEMGEN::DemandStream::generateValueOfTime ()

Generate the value of time.

Definition at line 531 of file DemandStream.cpp.

References \_demandCharacteristics, \_demandCharacteristicsRandomGenerator, TRADEMGEN::DemandCharacteristics::\_valueOfTimeCumulativeDistribution, and TRADEMGEN::ContinuousAttributeLite< T >::getValue().

Referenced by generateNextRequest().

#### 10.23.4.40 stdair::BookingRequestPtr\_T TRADEMGEN::DemandStream::generateNextRequest (stdair::RandomGeneration &, const stdair::DemandGenerationMethod &)

Generate the next request.

##### Parameters:

*stdair::RandomGeneration* Random generator.

*const stdair::DemandGenerationMethod::EN\_DemandGenerationMethod* Method used to generate the date time of the next booking request: statistic order or poisson process.

##### Returns:

stdair::BookingRequestPtr\_T Next request to be simulate.

Definition at line 541 of file DemandStream.cpp.

References TRADEMGEN::DemandCharacteristics::\_changeFeeDisutility, \_demandCharacteristics, \_key, TRADEMGEN::DemandCharacteristics::\_nonRefundableDisutility, describeKey(), generateChangeFees(), generateChannel(), generateFrequentFlyer(), generateNonRefundable(), generatePOS(), generatePreferredDepartureTime(), generateStayDuration(), generateTimeOfRequestPoissonProcess(), generateTimeOfRequestStatisticsOrder(), generateTripType(), generateValueOfTime(), generateWTP(), TRADEMGEN::DemandStreamKey::getDestination(), TRADEMGEN::DemandStreamKey::getOrigin(), TRADEMGEN::DemandStreamKey::getPreferredCabin(), and TRADEMGEN::DemandStreamKey::getPreferredDepartureDate().

#### 10.23.4.41 void TRADEMGEN::DemandStream::reset (stdair::BaseGenerator\_T & ioSharedGenerator)

Reset all the contexts of the demand stream.

Definition at line 625 of file DemandStream.cpp.

References \_randomGenerationContext, and TRADEMGEN::RandomGenerationContext::reset().

#### 10.23.4.42 void TRADEMGEN::DemandStream::toStream (std::ostream & ioOut) const [inline]

Dump a Business Object into an output stream.

##### Parameters:

*ostream&* the output stream.

Definition at line 292 of file DemandStream.hpp.

References toString().

#### 10.23.4.43 void TRADEMGEN::DemandStream::fromStream (std::istream & ioIn) [inline]

Read a Business Object from an input stream.

##### Parameters:

*istream&* the input stream.

Definition at line 300 of file DemandStream.hpp.



**10.23.4.44 std::string TRADEMGEN::DemandStream::toString () const**

Get the serialised version of the Business Object.

Definition at line 73 of file DemandStream.cpp.

References `_key`, and `TRADEMGEN::DemandStreamKey::toString()`.

Referenced by `toStream()`.

**10.23.4.45 const std::string TRADEMGEN::DemandStream::describeKey () const [inline]**

Get a string describing the key.

Definition at line 311 of file DemandStream.hpp.

References `_key`, and `TRADEMGEN::DemandStreamKey::toString()`.

Referenced by `generateNextRequest()`.

**10.23.4.46 std::string TRADEMGEN::DemandStream::display () const**

Dump recursively the content of the [DemandStream](#) object.

Definition at line 119 of file DemandStream.cpp.

References `_demandCharacteristics`, `_demandCharacteristicsRandomGenerator`, `_demandDistribution`, `_key`, `_posProMass`, `_randomGenerationContext`, `_requestDateTimeRandomGenerator`, `_totalNumberOfRequestsToBeGenerated`, `TRADEMGEN::DemandDistribution::describe()`, `TRADEMGEN::DemandCharacteristics::describe()`, `TRADEMGEN::CategoricalAttributeLite< T >::displayProbabilityMass()`, and `TRADEMGEN::DemandStreamKey::toString()`.

Referenced by `TRADEMGEN::BomDisplay::csvDisplay()`.

**10.23.4.47 const std::duration<T> TRADEMGEN::DemandStream::convertFloatIntoDuration (const std::duration<T>)**

Definition at line 401 of file DemandStream.cpp.

Referenced by `generateTimeOfRequestPoissonProcess()`, and `generateTimeOfRequestStatisticsOrder()`.

**10.23.5 Friends And Related Function Documentation****10.23.5.1 friend class std::duration<T> TRADEMGEN::FacBom [friend]**

Definition at line 31 of file DemandStream.hpp.

**10.23.5.2 friend class std::duration<T> TRADEMGEN::FacBomManager [friend]**

Definition at line 32 of file DemandStream.hpp.

**10.23.6 Member Data Documentation****10.23.6.1 Key\_T TRADEMGEN::DemandStream::\_key [protected]**

Primary key (string gathering the origin, destination, POS and date).

Definition at line 346 of file DemandStream.hpp.

Referenced by describeKey(), display(), generateNextRequest(), generateTimeOfRequestPoissonProcess(), generateTimeOfRequestStatisticsOrder(), getDestination(), getKey(), getOrigin(), getPreferredCabin(), getPreferredDepartureDate(), and toString().

#### 10.23.6.2 BomAbstract\* TRADEMGEN::DemandStream::\_parent [protected]

Pointer on the parent class (EventQueue).

Definition at line 351 of file DemandStream.hpp.

Referenced by getParent().

#### 10.23.6.3 stdair::HolderMap\_T TRADEMGEN::DemandStream::\_holderMap [protected]

Map holding the children (not used for now).

Definition at line 356 of file DemandStream.hpp.

Referenced by getHolderMap().

#### 10.23.6.4 DemandCharacteristics TRADEMGEN::DemandStream::\_demandCharacteristics [protected]

Demand characteristics.

Definition at line 361 of file DemandStream.hpp.

Referenced by display(), generateChangeFees(), generateChannel(), generateFrequentFlyer(), generateNextRequest(), generateNonRefundable(), generatePOS(), generatePreferredDepartureTime(), generateStayDuration(), generateTimeOfRequestPoissonProcess(), generateTimeOfRequestStatisticsOrder(), generateTripType(), generateValueOfTime(), generateWTP(), getChangeFeeDisutility(), getDemandCharacteristics(), getNonRefundableDisutility(), and setDemandCharacteristics().

#### 10.23.6.5 DemandDistribution TRADEMGEN::DemandStream::\_demandDistribution [protected]

Demand distribution.

Definition at line 366 of file DemandStream.hpp.

Referenced by display(), generateTimeOfRequestPoissonProcess(), getDemandDistribution(), getMeanNumberOfRequests(), getStdDevNumberOfRequests(), and setDemandDistribution().

#### 10.23.6.6 stdair::NbOfRequests\_T TRADEMGEN::DemandStream::\_totalNumberOfRequestsToBeGenerated [protected]

Total number of requests to be generated.

Definition at line 371 of file DemandStream.hpp.

Referenced by display(), generateTimeOfRequestStatisticsOrder(), getTotalNumberOfRequestsToBeGenerated(), setTotalNumberOfRequestsToBeGenerated(), and stillHavingRequestsToBeGenerated().

#### 10.23.6.7 RandomGenerationContext TRADEMGEN::DemandStream::\_randomGenerationContext [protected]

Random generation context.

Definition at line 376 of file DemandStream.hpp.

Referenced by `display()`, `generateTimeOfRequestStatisticsOrder()`, `getNumberOfRequestsGeneratedSoFar()`, `incrementGeneratedRequestsCounter()`, `reset()`, `setNumberOfRequestsGeneratedSoFar()`, and `stillHavingRequestsToBeGenerated()`.

#### 10.23.6.8 stdair::RandomGeneration TRADEMGEN::DemandStream::\_requestDateTimeRandomGenerator [protected]

Random generator for request date-time.

Definition at line 381 of file DemandStream.hpp.

Referenced by `display()`, `generateTimeOfRequestPoissonProcess()`, `generateTimeOfRequestStatisticsOrder()`, and `setRequestDateTimeRandomGeneratorSeed()`.

#### 10.23.6.9 stdair::RandomGeneration TRADEMGEN::DemandStream::\_demandCharacteristicsRandomGenerator [protected]

Random generator for demand characteristics.

Definition at line 386 of file DemandStream.hpp.

Referenced by `display()`, `generateChangeFees()`, `generateChannel()`, `generateFrequentFlyer()`, `generateNonRefundable()`, `generatePOS()`, `generatePreferredDepartureTime()`, `generateStayDuration()`, `generateTripType()`, `generateValueOfTime()`, and `setDemandCharacteristicsRandomGeneratorSeed()`.

#### 10.23.6.10 POSProbabilityMass\_T TRADEMGEN::DemandStream::\_posProMass [protected]

Default POS probability mass, used when "row" (rest of the world) is drawn.

Definition at line 392 of file DemandStream.hpp.

Referenced by `display()`, `getPOSProbabilityMass()`, and `setPOSProbabilityMass()`.

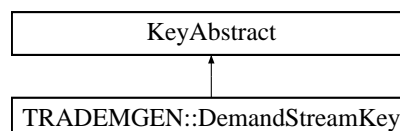
The documentation for this class was generated from the following files:

- trademgen/bom/[DemandStream.hpp](#)
- trademgen/bom/[DemandStream.cpp](#)

## 10.24 TRADEMGEN::DemandStreamKey Struct Reference

```
#include <trademgen/bom/DemandStreamKey.hpp>
```

Inheritance diagram for TRADEMGEN::DemandStreamKey::



## Public Member Functions

- [DemandStreamKey](#) (const stdair::AirportCode\_T &iOrigin, const stdair::AirportCode\_T &iDestination, const stdair::Date\_T &iPreferredDepartureDate, const stdair::CabinCode\_T &iPreferredCabin)
- [DemandStreamKey](#) (const [DemandStreamKey](#) &)
- [~DemandStreamKey](#) ()
- const stdair::AirportCode\_T & [getOrigin](#) () const
- const stdair::AirportCode\_T & [getDestination](#) () const
- const stdair::Date\_T & [getPreferredDepartureDate](#) () const
- const stdair::CabinCode\_T & [getPreferredCabin](#) () const
- void [toStream](#) (std::ostream &ioOut) const
- void [fromStream](#) (std::istream &ioIn)
- const std::string [toString](#) () const

### 10.24.1 Detailed Description

Key of a given demand-stream, made of a pair of required airports/cities (origin and destination), a preferred departure date and a preferred cabin. Those attributes correspond to a the travel requirements of a simulated traveller.

Definition at line 20 of file DemandStreamKey.hpp.

### 10.24.2 Constructor & Destructor Documentation

#### 10.24.2.1 TRADEMGEN::DemandStreamKey::DemandStreamKey (const stdair::AirportCode\_T &iOrigin, const stdair::AirportCode\_T &iDestination, const stdair::Date\_T &iPreferredDepartureDate, const stdair::CabinCode\_T &iPreferredCabin)

Constructor.

Definition at line 25 of file DemandStreamKey.cpp.

#### 10.24.2.2 TRADEMGEN::DemandStreamKey::DemandStreamKey (const [DemandStreamKey](#) &)

Default copy constructor.

Definition at line 35 of file DemandStreamKey.cpp.

#### 10.24.2.3 TRADEMGEN::DemandStreamKey::~~DemandStreamKey ()

Destructor.

Definition at line 42 of file DemandStreamKey.cpp.

### 10.24.3 Member Function Documentation

#### 10.24.3.1 const stdair::AirportCode\_T& TRADEMGEN::DemandStreamKey::getOrigin () const [inline]

Get the origin.

Definition at line 43 of file DemandStreamKey.hpp.

Referenced by TRADEMGEN::DemandStream::generateNextRequest(), and TRADEMGEN::DemandStream::getOrigin().

**10.24.3.2 const stdair::AirportCode\_T& TRADEMGEN::DemandStreamKey::getDestination () const [inline]**

Get the destination.

Definition at line 48 of file DemandStreamKey.hpp.

Referenced by TRADEMGEN::DemandStream::generateNextRequest(), and TRADEMGEN::DemandStream::getDestination().

**10.24.3.3 const stdair::Date\_T& TRADEMGEN::DemandStreamKey::getPreferredDepartureDate () const [inline]**

Get the preferred departure date.

Definition at line 53 of file DemandStreamKey.hpp.

Referenced by TRADEMGEN::DemandStream::generateNextRequest(), TRADEMGEN::DemandStream::generateTimeOfRequestPoissonProcess(), TRADEMGEN::DemandStream::generateTimeOfRequestStatisticsOrder(), and TRADEMGEN::DemandStream::getPreferredDepartureDate().

**10.24.3.4 const stdair::CabinCode\_T& TRADEMGEN::DemandStreamKey::getPreferredCabin () const [inline]**

Get the preferred cabin.

Definition at line 58 of file DemandStreamKey.hpp.

Referenced by TRADEMGEN::DemandStream::generateNextRequest(), and TRADEMGEN::DemandStream::getPreferredCabin().

**10.24.3.5 void TRADEMGEN::DemandStreamKey::toStream (std::ostream & ioOut) const**

Dump a Business Object Key into an output stream.

**Parameters:**

*ostream&* the output stream.

Definition at line 46 of file DemandStreamKey.cpp.

References toString().

**10.24.3.6 void TRADEMGEN::DemandStreamKey::fromStream (std::istream & ioIn)**

Read a Business Object Key from an input stream.

**Parameters:**

*istream&* the input stream.

Definition at line 51 of file DemandStreamKey.cpp.

### 10.24.3.7 const std::string TRADEMGEN::DemandStreamKey::toString () const

Get the serialised version of the Business Object Key.

That string is unique, at the level of a given Business Object, when among children of a given parent Business Object.

For instance, "H" and "K" allow to differentiate among two marketing classes for the same segment-stream.

Definition at line 55 of file DemandStreamKey.cpp.

Referenced by TRADEMGEN::DemandStream::describeKey(), TRADEMGEN::DemandStream::display(), toStream(), and TRADEMGEN::DemandStream::toString().

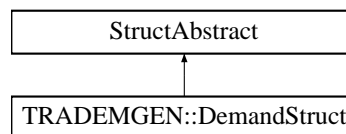
The documentation for this struct was generated from the following files:

- trademgen/bom/[DemandStreamKey.hpp](#)
- trademgen/bom/[DemandStreamKey.cpp](#)

## 10.25 TRADEMGEN::DemandStruct Struct Reference

```
#include <trademgen/bom/DemandStruct.hpp>
```

Inheritance diagram for TRADEMGEN::DemandStruct::



### Public Member Functions

- stdair::Date\_T [getDate](#) () const
- stdair::Duration\_T [getTime](#) () const
- const std::string [describe](#) () const
- [DemandStruct](#) ()
- [~DemandStruct](#) ()

### Public Attributes

- stdair::DatePeriod\_T [\\_dateRange](#)
- stdair::DoWStruct [\\_dow](#)
- stdair::AirportCode\_T [\\_origin](#)
- stdair::AirportCode\_T [\\_destination](#)
- stdair::CabinCode\_T [\\_prefCabin](#)
- stdair::MeanValue\_T [\\_demandMean](#)
- stdair::StdDevValue\_T [\\_demandStdDev](#)
- stdair::ChangeFeesRatio\_T [\\_changeFeeProb](#)
- stdair::Disutility\_T [\\_changeFeeDisutility](#)
- stdair::NonRefundableRatio\_T [\\_nonRefundableProb](#)
- stdair::Disutility\_T [\\_nonRefundableDisutility](#)
- [POSProbabilityMassFunction\\_T](#) [\\_posProbDist](#)

- [ChannelProbabilityMassFunction\\_T \\_channelProbDist](#)
- [TripTypeProbabilityMassFunction\\_T \\_tripProbDist](#)
- [StayDurationProbabilityMassFunction\\_T \\_stayProbDist](#)
- [FrequentFlyerProbabilityMassFunction\\_T \\_ffProbDist](#)
- [PreferredDepartureTimeContinuousDistribution\\_T \\_prefDepTimeProbDist](#)
- [stdair::WTP\\_T \\_minWTP](#)
- [ValueOfTimeContinuousDistribution\\_T \\_timeValueProbDist](#)
- [ArrivalPatternCumulativeDistribution\\_T \\_dtdProbDist](#)
- [stdair::Date\\_T \\_prefDepDateStart](#)
- [stdair::Date\\_T \\_prefDepDateEnd](#)
- [unsigned int \\_itYear](#)
- [unsigned int \\_itMonth](#)
- [unsigned int \\_itDay](#)
- [long \\_itHours](#)
- [long \\_itMinutes](#)
- [long \\_itSeconds](#)
- [stdair::AirportCode\\_T \\_itPosCode](#)
- [stdair::ChannelLabel\\_T \\_itChannelCode](#)
- [stdair::TripType\\_T \\_itTripCode](#)
- [stdair::DayDuration\\_T \\_itStayDuration](#)
- [stdair::FrequentFlyer\\_T \\_itFFCode](#)
- [stdair::Duration\\_T \\_itPrefDepTime](#)
- [stdair::PriceValue\\_T \\_itTimeValue](#)
- [stdair::DayDuration\\_T \\_itDTD](#)

### 10.25.1 Detailed Description

Utility Structure for the parsing of Demand structures.

Definition at line 21 of file DemandStruct.hpp.

### 10.25.2 Constructor & Destructor Documentation

#### 10.25.2.1 TRADEMGEN::DemandStruct::DemandStruct ()

Default constructor.

Definition at line 18 of file DemandStruct.cpp.

#### 10.25.2.2 TRADEMGEN::DemandStruct::~DemandStruct ()

Destructor

Definition at line 26 of file DemandStruct.cpp.

### 10.25.3 Member Function Documentation

#### 10.25.3.1 stdair::Date\_T TRADEMGEN::DemandStruct::getDate () const

Get the date from the staging details.

Definition at line 30 of file DemandStruct.cpp.

References `_itDay`, `_itMonth`, and `_itYear`.

Referenced by `TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator()()`, and `TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator()()`.

### 10.25.3.2 `stdair::Duration_T TRADEMGEN::DemandStruct::getTime () const`

Get the time from the staging details.

Definition at line 35 of file `DemandStruct.cpp`.

References `_itHours`, `_itMinutes`, and `_itSeconds`.

Referenced by `TRADEMGEN::DemandParserHelper::storePrefDepTime::operator()()`.

### 10.25.3.3 `const std::string TRADEMGEN::DemandStruct::describe () const`

Give a description of the structure (for display purposes).

Definition at line 42 of file `DemandStruct.cpp`.

References `_changeFeeProb`, `_channelProbDist`, `_dateRange`, `_demandMean`, `_demandStdDev`, `_destination`, `_dow`, `_dtdProbDist`, `_ffProbDist`, `_minWTP`, `_nonRefundableProb`, `_origin`, `_posProbDist`, `_prefCabin`, `_prefDepTimeProbDist`, `_stayProbDist`, `_timeValueProbDist`, and `_tripProbDist`.

## 10.25.4 Member Data Documentation

### 10.25.4.1 `stdair::DatePeriod_T TRADEMGEN::DemandStruct::_dateRange`

Definition at line 51 of file `DemandStruct.hpp`.

Referenced by `describe()`, and `TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator()()`.

### 10.25.4.2 `stdair::DoWStruct TRADEMGEN::DemandStruct::_dow`

Definition at line 52 of file `DemandStruct.hpp`.

Referenced by `describe()`, and `TRADEMGEN::DemandParserHelper::storeDow::operator()()`.

### 10.25.4.3 `stdair::AirportCode_T TRADEMGEN::DemandStruct::_origin`

Definition at line 53 of file `DemandStruct.hpp`.

Referenced by `describe()`, and `TRADEMGEN::DemandParserHelper::storeOrigin::operator()()`.

### 10.25.4.4 `stdair::AirportCode_T TRADEMGEN::DemandStruct::_destination`

Definition at line 54 of file `DemandStruct.hpp`.

Referenced by `describe()`, and `TRADEMGEN::DemandParserHelper::storeDestination::operator()()`.

### 10.25.4.5 `stdair::CabinCode_T TRADEMGEN::DemandStruct::_prefCabin`

Definition at line 55 of file `DemandStruct.hpp`.

Referenced by `describe()`, and `TRADEMGEN::DemandParserHelper::storePrefCabin::operator()()`.



**10.25.4.6 stdair::MeanValue\_T TRADEMGEN::DemandStruct::\_demandMean**

Definition at line 56 of file DemandStruct.hpp.

Referenced by describe(), and TRADEMGEN::DemandParserHelper::storeDemandMean::operator()().

**10.25.4.7 stdair::StdDevValue\_T TRADEMGEN::DemandStruct::\_demandStdDev**

Definition at line 57 of file DemandStruct.hpp.

Referenced by describe(), and TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator()().

**10.25.4.8 stdair::ChangeFeesRatio\_T TRADEMGEN::DemandStruct::\_changeFeeProb**

Definition at line 58 of file DemandStruct.hpp.

Referenced by describe(), and TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator()().

**10.25.4.9 stdair::Disutility\_T TRADEMGEN::DemandStruct::\_changeFeeDisutility**

Definition at line 59 of file DemandStruct.hpp.

Referenced by TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator()().

**10.25.4.10 stdair::NonRefundableRatio\_T TRADEMGEN::DemandStruct::\_nonRefundableProb**

Definition at line 60 of file DemandStruct.hpp.

Referenced by describe(), and TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator()().

**10.25.4.11 stdair::Disutility\_T TRADEMGEN::DemandStruct::\_nonRefundableDisutility**

Definition at line 61 of file DemandStruct.hpp.

Referenced by TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator()().

**10.25.4.12 POSProbabilityMassFunction\_T TRADEMGEN::DemandStruct::\_posProbDist**

Definition at line 62 of file DemandStruct.hpp.

Referenced by describe(), TRADEMGEN::DemandParserHelper::doEndDemand::operator()(), and TRADEMGEN::DemandParserHelper::storePosProbMass::operator()().

**10.25.4.13 ChannelProbabilityMassFunction\_T TRADEMGEN::DemandStruct::\_channelProbDist**

Definition at line 63 of file DemandStruct.hpp.

Referenced by describe(), TRADEMGEN::DemandParserHelper::doEndDemand::operator()(), and TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator()().

**10.25.4.14 TripTypeProbabilityMassFunction\_T TRADEMGEN::DemandStruct::\_tripProbDist**

Definition at line 64 of file DemandStruct.hpp.

Referenced by describe(), TRADEMGEN::DemandParserHelper::doEndDemand::operator>(), and TRADEMGEN::DemandParserHelper::storeTripProbMass::operator().

**10.25.4.15 StayDurationProbabilityMassFunction\_T TRADEMGEN::DemandStruct::\_stayProbDist**

Definition at line 65 of file DemandStruct.hpp.

Referenced by describe(), TRADEMGEN::DemandParserHelper::doEndDemand::operator>(), and TRADEMGEN::DemandParserHelper::storeStayProbMass::operator().

**10.25.4.16 FrequentFlyerProbabilityMassFunction\_T TRADEMGEN::DemandStruct::\_ffProbDist**

Definition at line 66 of file DemandStruct.hpp.

Referenced by describe(), TRADEMGEN::DemandParserHelper::doEndDemand::operator>(), and TRADEMGEN::DemandParserHelper::storeFFProbMass::operator().

**10.25.4.17 PreferredDepartureTimeContinuousDistribution\_T TRADEMGEN::DemandStruct::\_prefDepTimeProbDist**

Definition at line 67 of file DemandStruct.hpp.

Referenced by describe(), TRADEMGEN::DemandParserHelper::doEndDemand::operator>(), and TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator().

**10.25.4.18 stdair::WTP\_T TRADEMGEN::DemandStruct::\_minWTP**

Definition at line 68 of file DemandStruct.hpp.

Referenced by describe(), and TRADEMGEN::DemandParserHelper::storeWTP::operator().

**10.25.4.19 ValueOfTimeContinuousDistribution\_T TRADEMGEN::DemandStruct::\_timeValueProbDist**

Definition at line 69 of file DemandStruct.hpp.

Referenced by describe(), TRADEMGEN::DemandParserHelper::doEndDemand::operator>(), and TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator().

**10.25.4.20 ArrivalPatternCumulativeDistribution\_T TRADEMGEN::DemandStruct::\_dtdProbDist**

Definition at line 70 of file DemandStruct.hpp.

Referenced by describe(), TRADEMGEN::DemandParserHelper::doEndDemand::operator>(), and TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator().

**10.25.4.21 stdair::Date\_T TRADEMGEN::DemandStruct::\_prefDepDateStart**

Staging Date.

Definition at line 75 of file DemandStruct.hpp.

Referenced by TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator>(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

#### 10.25.4.22 stdair::Date\_T TRADEMGEN::DemandStruct::\_prefDepDateEnd

Definition at line 76 of file DemandStruct.hpp.

Referenced by TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator().

#### 10.25.4.23 unsigned int TRADEMGEN::DemandStruct::\_itYear

Definition at line 77 of file DemandStruct.hpp.

Referenced by getDate().

#### 10.25.4.24 unsigned int TRADEMGEN::DemandStruct::\_itMonth

Definition at line 78 of file DemandStruct.hpp.

Referenced by getDate().

#### 10.25.4.25 unsigned int TRADEMGEN::DemandStruct::\_itDay

Definition at line 79 of file DemandStruct.hpp.

Referenced by getDate().

#### 10.25.4.26 long TRADEMGEN::DemandStruct::\_itHours

Staging Time.

Definition at line 82 of file DemandStruct.hpp.

Referenced by getTime().

#### 10.25.4.27 long TRADEMGEN::DemandStruct::\_itMinutes

Definition at line 83 of file DemandStruct.hpp.

Referenced by getTime(), and TRADEMGEN::DemandParserHelper::storePrefDepTime::operator().

#### 10.25.4.28 long TRADEMGEN::DemandStruct::\_itSeconds

Definition at line 84 of file DemandStruct.hpp.

Referenced by getTime(), TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

#### 10.25.4.29 stdair::AirportCode\_T TRADEMGEN::DemandStruct::\_itPosCode

Staging Point-Of-Sale (POS) code.

Definition at line 87 of file DemandStruct.hpp.

Referenced by TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), and TRADEMGEN::DemandParserHelper::storePosCode::operator().

**10.25.4.30 stdair::ChannelLabel\_T TRADEMGEN::DemandStruct::\_itChannelCode**

Staging channel type code.

Definition at line 90 of file DemandStruct.hpp.

Referenced by TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), and TRADEMGEN::DemandParserHelper::storeChannelCode::operator().

**10.25.4.31 stdair::TripType\_T TRADEMGEN::DemandStruct::\_itTripCode**

Staging trip type code.

Definition at line 93 of file DemandStruct.hpp.

Referenced by TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), and TRADEMGEN::DemandParserHelper::storeTripCode::operator().

**10.25.4.32 stdair::DayDuration\_T TRADEMGEN::DemandStruct::\_itStayDuration**

Staging stay duration.

Definition at line 96 of file DemandStruct.hpp.

Referenced by TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), and TRADEMGEN::DemandParserHelper::storeStayCode::operator().

**10.25.4.33 stdair::FrequentFlyer\_T TRADEMGEN::DemandStruct::\_itFFCode**

Staging Frequent Flyer code.

Definition at line 99 of file DemandStruct.hpp.

Referenced by TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), and TRADEMGEN::DemandParserHelper::storeFFCode::operator().

**10.25.4.34 stdair::Duration\_T TRADEMGEN::DemandStruct::\_itPrefDepTime**

Staging preferred departure time.

Definition at line 102 of file DemandStruct.hpp.

Referenced by TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepTime::operator().

**10.25.4.35 stdair::PriceValue\_T TRADEMGEN::DemandStruct::\_itTimeValue**

Staging time value.

Definition at line 105 of file DemandStruct.hpp.

Referenced by TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator(), and TRADEMGEN::DemandParserHelper::storeTimeValue::operator().

**10.25.4.36 stdair::DayDuration\_T TRADEMGEN::DemandStruct::\_itDTD**

Staging DTD (Days-To-Departure).

Definition at line 108 of file DemandStruct.hpp.

Referenced by TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), and TRADEMGEN::DemandParserHelper::storeDTD::operator().

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

- trademgen/bom/[DemandStruct.hpp](#)
- trademgen/bom/[DemandStruct.cpp](#)

## 10.26 TRADEMGEN::DictionaryManager Class Reference

Class wrapper of dictionary business methods.

```
#include <trademgen/basic/DictionaryManager.hpp>
```

### Static Public Member Functions

- static const stdair::Probability\_T [keyToValue](#) (const [DictionaryKey\\_T](#))
- static const [DictionaryKey\\_T](#) [valueToKey](#) (const stdair::Probability\_T)

### 10.26.1 Detailed Description

Class wrapper of dictionary business methods.

Definition at line 21 of file DictionaryManager.hpp.

### 10.26.2 Member Function Documentation

#### 10.26.2.1 const stdair::Probability\_T TRADEMGEN::DictionaryManager::keyToValue (const *DictionaryKey\_T*) [static]

Convert from key to value.

Definition at line 10 of file DictionaryManager.cpp.

Referenced by TRADEMGEN::ContinuousAttributeLite< T >::displayCumulativeDistribution(), TRADEMGEN::ContinuousAttribute< T >::displayCumulativeDistribution(), TRADEMGEN::ContinuousAttribute< T >::displayInverseCumulativeDistribution(), TRADEMGEN::CategoricalAttributeLite< T >::displayProbabilityMass(), TRADEMGEN::ContinuousAttributeLite< T >::getDerivativeValue(), TRADEMGEN::ContinuousAttributeLite< T >::getValue(), and TRADEMGEN::ContinuousAttribute< T >::getValue().

#### 10.26.2.2 const [DictionaryKey\\_T](#) TRADEMGEN::DictionaryManager::valueToKey (const stdair::Probability\_T) [static]

Convert from value to key.

Definition at line 17 of file DictionaryManager.cpp.

Referenced by TRADEMGEN::ContinuousAttributeLite< T >::getValue(), TRADEMGEN::ContinuousAttribute< T >::getValue(), and TRADEMGEN::CategoricalAttributeLite< T >::getValue().

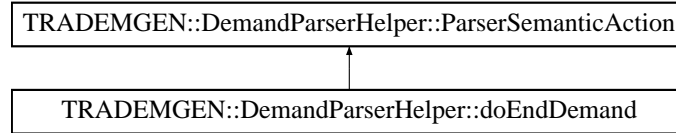
The documentation for this class was generated from the following files:

- trademgen/basic/[DictionaryManager.hpp](#)
- trademgen/basic/[DictionaryManager.cpp](#)

## 10.27 TRADEMGEN::DemandParserHelper::doEndDemand Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::doEndDemand::



### Public Member Functions

- `doEndDemand` (SEVMGR::SEVMGR\_ServicePtr\_T, stdair::RandomGeneration &, const POSProbabilityMass\_T &, DemandStruct &)
- void `operator()` (iterator\_t iStr, iterator\_t iStrEnd) const

### Public Attributes

- SEVMGR::SEVMGR\_ServicePtr\_T `_sevmgrServicePtr`
- stdair::RandomGeneration & `_uniformGenerator`
- const POSProbabilityMass\_T & `_posProbabilityMass`
- DemandStruct & `_demand`

#### 10.27.1 Detailed Description

Mark the end of the demand parsing.

Definition at line 275 of file DemandParserHelper.hpp.

#### 10.27.2 Constructor & Destructor Documentation

##### 10.27.2.1 TRADEMGEN::DemandParserHelper::doEndDemand::doEndDemand (SEVMGR::SEVMGR\_ServicePtr\_T, stdair::RandomGeneration &, const POSProbabilityMass\_T &, DemandStruct &)

Actor Constructor.

Definition at line 457 of file DemandParserHelper.cpp.

#### 10.27.3 Member Function Documentation

##### 10.27.3.1 void TRADEMGEN::DemandParserHelper::doEndDemand::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 469 of file DemandParserHelper.cpp.

References TRADEMGEN::DemandStruct::\_channelProbDist, TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand, TRADEMGEN::DemandStruct::\_dtdProbDist,

TRADEMGENT::DemandStruct::\_ffProbDist, \_posProbabilityMass, TRADEMGENT::DemandStruct::\_posProbDist, TRADEMGENT::DemandStruct::\_prefDepTimeProbDist, \_sevmgrServicePtr, TRADEMGENT::DemandStruct::\_stayProbDist, TRADEMGENT::DemandStruct::\_timeValueProbDist, TRADEMGENT::DemandStruct::\_tripProbDist, and \_uniformGenerator.

#### 10.27.4 Member Data Documentation

##### 10.27.4.1 SEVMGR::SEVMGR\_ServicePtr\_T TRADEMGENT::DemandParserHelper::doEndDemand::\_sevmgrServicePtr

Actor Specific Context.

Definition at line 282 of file DemandParserHelper.hpp.

Referenced by operator().

##### 10.27.4.2 stdair::RandomGeneration& TRADEMGENT::DemandParserHelper::doEndDemand::\_uniformGenerator

Definition at line 283 of file DemandParserHelper.hpp.

Referenced by operator().

##### 10.27.4.3 const POSProbabilityMass\_T& TRADEMGENT::DemandParserHelper::doEndDemand::\_posProbabilityMass

Definition at line 284 of file DemandParserHelper.hpp.

Referenced by operator().

##### 10.27.4.4 DemandStruct& TRADEMGENT::DemandParserHelper::ParserSemanticAction::\_demand [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by operator(), TRADEMGENT::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGENT::DemandParserHelper::storeDTD::operator(), TRADEMGENT::DemandParserHelper::storeTimeValueProbMass::operator(), TRADEMGENT::DemandParserHelper::storeTimeValue::operator(), TRADEMGENT::DemandParserHelper::storeWTP::operator(), TRADEMGENT::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGENT::DemandParserHelper::storePrefDepTime::operator(), TRADEMGENT::DemandParserHelper::storeFFProbMass::operator(), TRADEMGENT::DemandParserHelper::storeFFCode::operator(), TRADEMGENT::DemandParserHelper::storeStayProbMass::operator(), TRADEMGENT::DemandParserHelper::storeStayCode::operator(), TRADEMGENT::DemandParserHelper::storeTripProbMass::operator(), TRADEMGENT::DemandParserHelper::storeTripCode::operator(), TRADEMGENT::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGENT::DemandParserHelper::storeChannelCode::operator(), TRADEMGENT::DemandParserHelper::storePosProbMass::operator(), TRADEMGENT::DemandParserHelper::storePosCode::operator(), TRADEMGENT::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGENT::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGENT::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGENT::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGENT::DemandParserHelper::storeDemandStdDev::operator(),

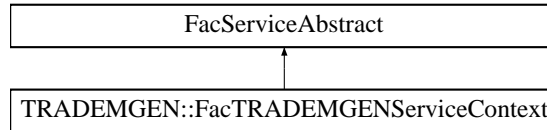
TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

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

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.28 FacServiceAbstract Class Reference

Inheritance diagram for FacServiceAbstract::



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

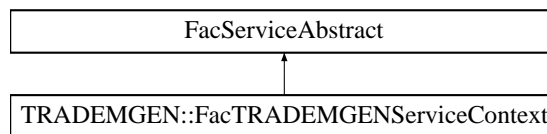
- trademgen/factory/[FacTRADEMGENServiceContext.hpp](#)

## 10.29 TRADEMGEN::FacTRADEMGENServiceContext Class Reference

Factory for creating the TraDemGen service context instance.

```
#include <trademgen/factory/FacTRADEMGENServiceContext.hpp>
```

Inheritance diagram for TRADEMGEN::FacTRADEMGENServiceContext::



### Public Member Functions

- [~FacTRADEMGENServiceContext \(\)](#)
- [TRADEMGEN\\_ServiceContext & create](#) (const stdair::RandomSeed\_T &)

### Static Public Member Functions

- static [FacTRADEMGENServiceContext & instance](#) ()



## Protected Member Functions

- [FacTRADEMGENSEerviceContext \(\)](#)

### 10.29.1 Detailed Description

Factory for creating the TraDemGen service context instance.

Definition at line 21 of file FacTRADEMGENSEerviceContext.hpp.

### 10.29.2 Constructor & Destructor Documentation

#### 10.29.2.1 TRADEMGEN::FacTRADEMGENSEerviceContext::~~FacTRADEMGENSEerviceContext ()

Destructor.

The Destruction put the `_instance` to NULL in order to be clean for the next [FacTRADEMGENSEerviceContext::instance\(\)](#).

Definition at line 17 of file FacTRADEMGENSEerviceContext.cpp.

#### 10.29.2.2 TRADEMGEN::FacTRADEMGENSEerviceContext::FacTRADEMGENSEerviceContext () [inline, protected]

Default Constructor.

This constructor is protected in order to ensure the singleton pattern.

Definition at line 57 of file FacTRADEMGENSEerviceContext.hpp.

Referenced by `instance()`.

### 10.29.3 Member Function Documentation

#### 10.29.3.1 [FacTRADEMGENSEerviceContext](#) & TRADEMGEN::FacTRADEMGENSEerviceContext::instance () [static]

Provide the unique instance.

The singleton is instantiated when first used.

#### Returns:

[FacTRADEMGENSEerviceContext](#)&

Definition at line 22 of file FacTRADEMGENSEerviceContext.cpp.

References `FacTRADEMGENSEerviceContext()`.

#### 10.29.3.2 [TRADEMGEN\\_ServiceContext](#) & TRADEMGEN::FacTRADEMGENSEerviceContext::create (const stdair::RandomSeed\_T &)

Create a new [TRADEMGEN\\_ServiceContext](#) object.

This new object is added to the list of instantiated objects.

**Parameters:**

**const** stdair::RandomSeed\_T& Seed for the random generation.

**Returns:**

[TRADEMGEN\\_ServiceContext](#)& The newly created object.

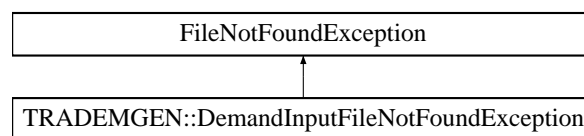
Definition at line 35 of file FacTRADEMGENServiceContext.cpp.

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

- trademgen/factory/[FacTRADEMGENServiceContext.hpp](#)
- trademgen/factory/[FacTRADEMGENServiceContext.cpp](#)

**10.30 FileNotFoundException Class Reference**

Inheritance diagram for FileNotFoundException::



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

- trademgen/[TRADEMGEN\\_Exceptions.hpp](#)

**10.31 TRADEMGEN::FlagSaver Struct Reference****Public Member Functions**

- [FlagSaver](#) (std::ostream &oStream)
- [~FlagSaver](#) ()

**10.31.1 Detailed Description**

Helper singleton structure to store the current formatting flags of any given output stream. The flags are re-set at the structure destruction.

Definition at line 24 of file BomDisplay.cpp.

**10.31.2 Constructor & Destructor Documentation****10.31.2.1 TRADEMGEN::FlagSaver::FlagSaver (std::ostream &oStream) [inline]**

Constructor.

Definition at line 27 of file BomDisplay.cpp.

**10.31.2.2 TRADEMGEN::FlagSaver::~FlagSaver () [inline]**

Destructor.

Definition at line 32 of file BomDisplay.cpp.

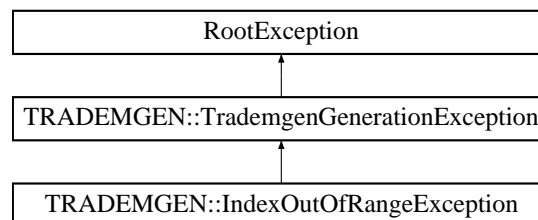
The documentation for this struct was generated from the following file:

- trademgen/bom/[BomDisplay.cpp](#)

**10.32 TRADEMGEN::IndexOutOfRangeException Class Reference**

```
#include <trademgen/TRADEMGEN_Exceptions.hpp>
```

Inheritance diagram for TRADEMGEN::IndexOutOfRangeException::

**Public Member Functions**

- [IndexOutOfRangeException](#) (const std::string &iWhat)

**10.32.1 Detailed Description**

Exception when index out of range

Definition at line 43 of file TRADEMGEN\_Exceptions.hpp.

**10.32.2 Constructor & Destructor Documentation****10.32.2.1 TRADEMGEN::IndexOutOfRangeException::IndexOutOfRangeException (const std::string &iWhat) [inline]**

Constructor.

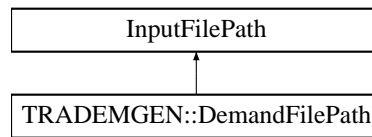
Definition at line 48 of file TRADEMGEN\_Exceptions.hpp.

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

- trademgen/[TRADEMGEN\\_Exceptions.hpp](#)

**10.33 InputFilePath Class Reference**

Inheritance diagram for InputFilePath::

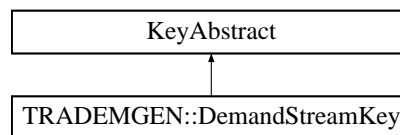


The documentation for this class was generated from the following file:

- trademgen/[TRADEMGEN\\_Types.hpp](#)

## 10.34 KeyAbstract Class Reference

Inheritance diagram for KeyAbstract::



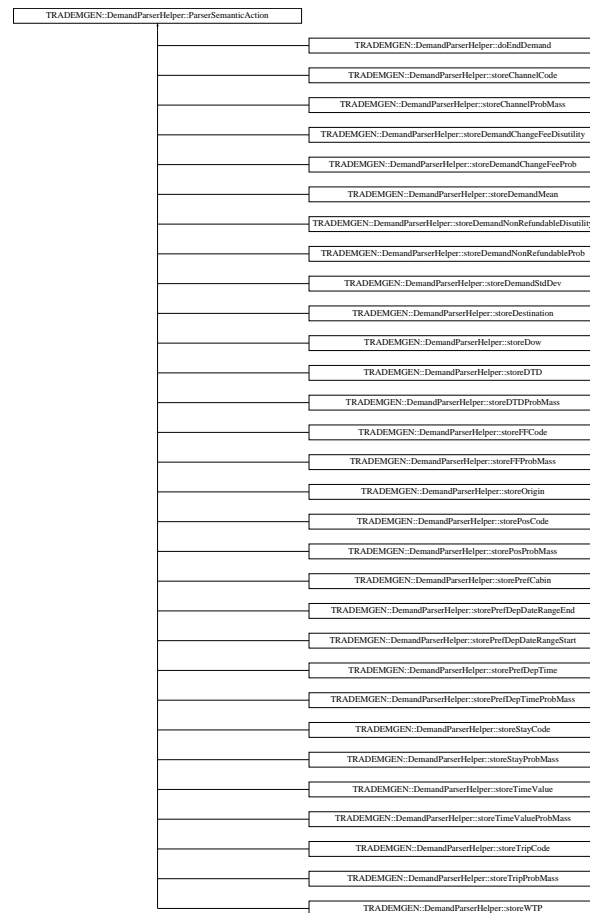
The documentation for this class was generated from the following file:

- trademgen/bom/[DemandStreamKey.hpp](#)

## 10.35 TRADEMGEN::DemandParserHelper::ParserSemanticAction Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::ParserSemanticAction::



## Public Member Functions

- [ParserSemanticAction](#) ([DemandStruct](#) &)

## Public Attributes

- [DemandStruct](#) & [\\_demand](#)

### 10.35.1 Detailed Description

Generic Semantic Action (Actor / Functor) for the Demand Parser.

Definition at line 31 of file DemandParserHelper.hpp.

### 10.35.2 Constructor & Destructor Documentation

#### 10.35.2.1 TRADEMGEN::DemandParserHelper::ParserSemanticAction::ParserSemanticAction ([DemandStruct](#) &)

Actor Constructor.

Definition at line 26 of file DemandParserHelper.cpp.

### 10.35.3 Member Data Documentation

#### 10.35.3.1 DemandStruct& TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_-demand

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(), TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGEN::DemandParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(), TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(), TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

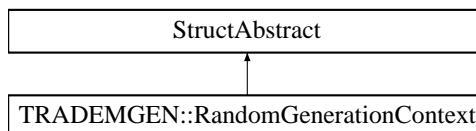
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.36 TRADEMGEN::RandomGenerationContext Struct Reference

```
#include <trademgen/basic/RandomGenerationContext.hpp>
```

Inheritance diagram for TRADEMGEN::RandomGenerationContext::



### Public Member Functions

- const stdair::Count\_T & [getNumberOfRequestsGeneratedSoFar](#) () const

- const stdair::Probability\_T & [getCumulativeProbabilitySoFar](#) () const
- void [setNumberOfRequestsGeneratedSoFar](#) (const stdair::Count\_T &iCount)
- void [setCumulativeProbabilitySoFar](#) (const stdair::Probability\_T &iProb)
- [RandomGenerationContext](#) ()
- [RandomGenerationContext](#) (const [RandomGenerationContext](#) &)
- [~RandomGenerationContext](#) ()
- void [incrementGeneratedRequestsCounter](#) ()
- void [reset](#) ()
- const std::string [describe](#) () const

### 10.36.1 Detailed Description

Structure holding the context necessary for demand random generation.

Definition at line 20 of file RandomGenerationContext.hpp.

### 10.36.2 Constructor & Destructor Documentation

#### 10.36.2.1 TRADEMGEN::RandomGenerationContext::RandomGenerationContext ()

Default constructor.

Definition at line 13 of file RandomGenerationContext.cpp.

#### 10.36.2.2 TRADEMGEN::RandomGenerationContext::RandomGenerationContext (const [RandomGenerationContext](#) &)

Default constructors.

Definition at line 20 of file RandomGenerationContext.cpp.

#### 10.36.2.3 TRADEMGEN::RandomGenerationContext::~~RandomGenerationContext ()

Destructor.

Definition at line 26 of file RandomGenerationContext.cpp.

### 10.36.3 Member Function Documentation

#### 10.36.3.1 const stdair::Count\_T& TRADEMGEN::RandomGenerationContext::getNumberOfRequestsGeneratedSoFar () const [inline]

Get the number of requests generated so far.

Definition at line 26 of file RandomGenerationContext.hpp.

Referenced by `TRADEMGENT::DemandStream::generateTimeOfRequestStatisticsOrder()`, `TRADEMGENT::DemandStream::getNumberOfRequestsGeneratedSoFar()`, and `TRADEMGENT::DemandStream::stillHavingRequestsToBeGenerated()`.

#### 10.36.3.2 const stdair::Probability\_T& TRADEMGEN::RandomGenerationContext::getCumulativeProbabilitySoFar () const [inline]

Get the cumulative probability in arrival pattern for last request generated so far (needed for sequential generation).

Definition at line 34 of file RandomGenerationContext.hpp.

Referenced by TRADEMGEN::DemandStream::generateTimeOfRequestStatisticsOrder().

#### 10.36.3.3 void TRADEMGEN::RandomGenerationContext::setNumberOfRequestsGeneratedSoFar (const std::count\_T & iCount) [inline]

Set the number of requests generated so far.

Definition at line 43 of file RandomGenerationContext.hpp.

Referenced by TRADEMGEN::DemandStream::setNumberOfRequestsGeneratedSoFar().

#### 10.36.3.4 void TRADEMGEN::RandomGenerationContext::setCumulativeProbabilitySoFar (const std::probability\_T & iProb) [inline]

Set the cumulative probability in arrival pattern for last request generated so far (needed for sequential generation).

Definition at line 51 of file RandomGenerationContext.hpp.

Referenced by TRADEMGEN::DemandStream::generateTimeOfRequestStatisticsOrder().

#### 10.36.3.5 void TRADEMGEN::RandomGenerationContext::incrementGeneratedRequestsCounter ()

Increment counter of requests generated so far.

Definition at line 38 of file RandomGenerationContext.cpp.

Referenced by TRADEMGEN::DemandStream::incrementGeneratedRequestsCounter().

#### 10.36.3.6 void TRADEMGEN::RandomGenerationContext::reset ()

Reset the counters.

Definition at line 43 of file RandomGenerationContext.cpp.

Referenced by TRADEMGEN::DemandStream::reset().

#### 10.36.3.7 const std::string TRADEMGEN::RandomGenerationContext::describe () const

Give a description of the structure (for display purposes).

Definition at line 30 of file RandomGenerationContext.cpp.

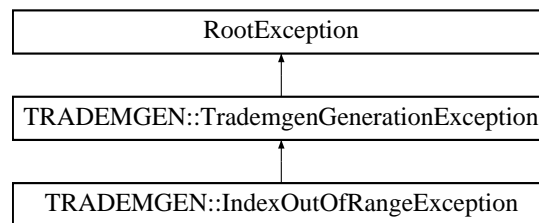
The documentation for this struct was generated from the following files:

- trademgen/basic/[RandomGenerationContext.hpp](#)
- trademgen/basic/[RandomGenerationContext.cpp](#)

## 10.37 RootException Class Reference

Inheritance diagram for RootException::



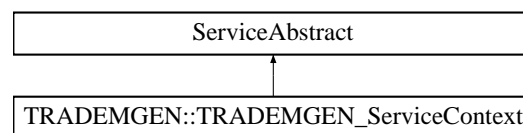


The documentation for this class was generated from the following file:

- trademgen/[TRADEMGEN\\_Exceptions.hpp](#)

## 10.38 ServiceAbstract Class Reference

Inheritance diagram for ServiceAbstract::



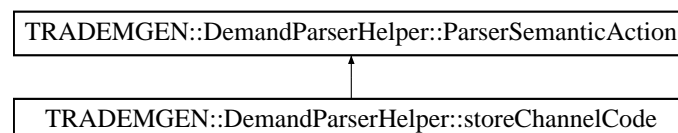
The documentation for this class was generated from the following file:

- trademgen/service/[TRADEMGEN\\_ServiceContext.hpp](#)

## 10.39 TRADEMGEN::DemandParserHelper::storeChannelCode Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeChannelCode::



### Public Member Functions

- [storeChannelCode](#) ([DemandStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

### 10.39.1 Detailed Description

Store the channel type code.

Definition at line 151 of file DemandParserHelper.hpp.

### 10.39.2 Constructor & Destructor Documentation

#### 10.39.2.1 TRADEMGEN::DemandParserHelper::storeChannelCode::storeChannelCode (DemandStruct &)

Actor Constructor.

Definition at line 219 of file DemandParserHelper.cpp.

### 10.39.3 Member Function Documentation

#### 10.39.3.1 void TRADEMGEN::DemandParserHelper::storeChannelCode::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 224 of file DemandParserHelper.cpp.

References TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand, and TRADEMGEN::DemandStruct::\_itChannelCode.

### 10.39.4 Member Data Documentation

#### 10.39.4.1 DemandStruct& TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(), TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGEN::DemandParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(), TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(), TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::store-

Destination::operator>(), TRADEMGEN::DemandParserHelper::storeOrigin::operator>(),  
 TRADEMGEN::DemandParserHelper::storeDow::operator()), TRADEMGEN::DemandParser-  
 Helper::storePrefDepDateRangeEnd::operator()), and TRADEMGEN::DemandParserHelper::store-  
 PrefDepDateRangeStart::operator()).

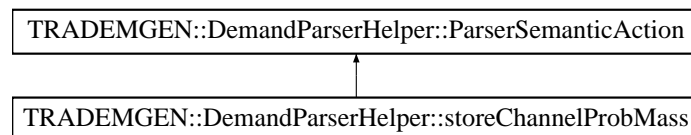
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.40 TRADEMGEN::DemandParserHelper::storeChannelProbMass Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeChannelProbMass::



### Public Member Functions

- [storeChannelProbMass](#) ([DemandStruct](#) &)
- void [operator\(\)](#) (double iReal) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

#### 10.40.1 Detailed Description

Store the channel type probability mass.

Definition at line 159 of file DemandParserHelper.hpp.

#### 10.40.2 Constructor & Destructor Documentation

##### 10.40.2.1 TRADEMGEN::DemandParserHelper::storeChannelProbMass::storeChannelProbMass ([DemandStruct](#) &)

Actor Constructor.

Definition at line 231 of file DemandParserHelper.cpp.

#### 10.40.3 Member Function Documentation

##### 10.40.3.1 void TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator() (double *iReal*) const

## 10.41 TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility Struct Reference 107

Actor Function (functor).

Definition at line 236 of file DemandParserHelper.cpp.

References TRADEMGEN::DemandStruct::\_channelProbDist, TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand, and TRADEMGEN::DemandStruct::\_itChannelCode.

### 10.40.4 Member Data Documentation

#### 10.40.4.1 DemandStruct & TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(), TRADEMGEN::DemandParserHelper::storeDTPProbMass::operator(), TRADEMGEN::DemandParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(), TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(), TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

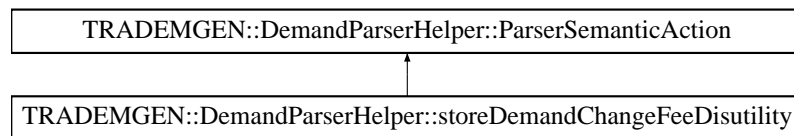
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.41 TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::



## Public Member Functions

- [storeDemandChangeFeeDisutility](#) ([DemandStruct](#) &)
- [void operator\(\)](#) (double *iReal*) const

## Public Attributes

- [DemandStruct](#) & [\\_demand](#)

### 10.41.1 Detailed Description

Store the demand change fee disutility.

Definition at line 111 of file DemandParserHelper.hpp.

### 10.41.2 Constructor & Destructor Documentation

#### 10.41.2.1 TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::storeDemandChangeFeeDisutility ([DemandStruct](#) &)

Actor Constructor.

Definition at line 150 of file DemandParserHelper.cpp.

### 10.41.3 Member Function Documentation

#### 10.41.3.1 void TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 155 of file DemandParserHelper.cpp.

References [TRADEMGEN::DemandStruct::\\_changeFeeDisutility](#), and [TRADEMGEN::DemandParserHelper::ParserSemanticAction::\\_demand](#).

### 10.41.4 Member Data Documentation

#### 10.41.4.1 [DemandStruct](#)& [TRADEMGEN::DemandParserHelper::ParserSemanticAction::\\_demand](#) [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by [TRADEMGEN::DemandParserHelper::doEndDemand::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeDTD::operator\(\)](#), and [TRADEMGEN::DemandParserHelper::storeTimeValue::operator\(\)](#).

ProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(),  
 TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParser-  
 Helper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::store-  
 PrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(),  
 TRADEMGEN::DemandParserHelper::storeFFCode::operator(), TRADEMGEN::Demand-  
 ParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::store-  
 StayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(),  
 TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::Demand-  
 ParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::store-  
 ChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(),  
 TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::Demand-  
 ParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParser-  
 Helper::storeDemandNonRefundableProb::operator(), operator(), TRADEMGEN::DemandParser-  
 Helper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::store-  
 DemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(),  
 TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParser-  
 Helper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(),  
 TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParser-  
 Helper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::store-  
 PrefDepDateRangeStart::operator().

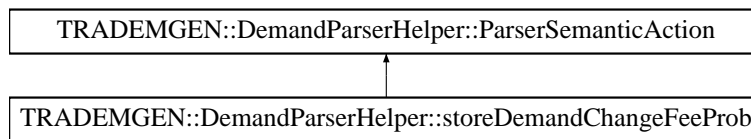
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.42 TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb:



### Public Member Functions

- [storeDemandChangeFeeProb](#) ([DemandStruct](#) &)
- [operator\(\)](#) (double iReal) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

#### 10.42.1 Detailed Description

Store the demand change fee probability.

Definition at line 103 of file DemandParserHelper.hpp.

## **10.42.2 Constructor & Destructor Documentation**

### **10.42.2.1 TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::storeDemandChangeFeeProb (DemandStruct &)**

Actor Constructor.

Definition at line 139 of file DemandParserHelper.cpp.

## **10.42.3 Member Function Documentation**

### **10.42.3.1 void TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator()(double *iReal*) const**

Actor Function (functor).

Definition at line 144 of file DemandParserHelper.cpp.

References TRADEMGEN::DemandStruct::\_changeFeeProb, and TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand.

## **10.42.4 Member Data Documentation**

### **10.42.4.1 DemandStruct& TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand** [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(), TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGEN::DemandParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(), TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(), TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

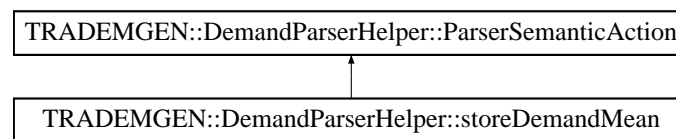
The documentation for this struct was generated from the following files:

- [trademgen/command/DemandParserHelper.hpp](#)
- [trademgen/command/DemandParserHelper.cpp](#)

## 10.43 TRADEMGEN::DemandParserHelper::storeDemandMean Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeDemandMean::



### Public Member Functions

- [storeDemandMean](#) ([DemandStruct](#) &)
- [void operator\(\)](#) (double iReal) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

#### 10.43.1 Detailed Description

Store the demand mean value.

Definition at line 87 of file DemandParserHelper.hpp.

#### 10.43.2 Constructor & Destructor Documentation

##### 10.43.2.1 TRADEMGEN::DemandParserHelper::storeDemandMean::storeDemandMean ([DemandStruct](#) &)

Actor Constructor.

Definition at line 117 of file DemandParserHelper.cpp.

#### 10.43.3 Member Function Documentation

##### 10.43.3.1 void TRADEMGEN::DemandParserHelper::storeDemandMean::operator() (double *i-Real*) const

Actor Function (functor).

Definition at line 122 of file DemandParserHelper.cpp.

References [TRADEMG...DemandParserHelper::ParserSemanticAction::\\_demand](#), and [TRADEMG...DemandStruct::\\_demandMean](#).



### 10.43.4 Member Data Documentation

#### 10.43.4.1 DemandStruct& TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(), TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGEN::DemandParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(), TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(), TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

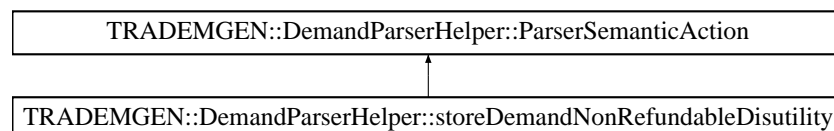
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

### 10.44 TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::



#### Public Member Functions

- [storeDemandNonRefundableDisutility](#) ([DemandStruct](#) &)

- void [operator\(\)](#) (double iReal) const

## Public Attributes

- [DemandStruct](#) & [\\_demand](#)

### 10.44.1 Detailed Description

Store the demand non refundable disutility.

Definition at line 127 of file DemandParserHelper.hpp.

### 10.44.2 Constructor & Destructor Documentation

#### 10.44.2.1 TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::storeDemandNonRefundableDisutility ([DemandStruct](#) &)

Actor Constructor.

Definition at line 174 of file DemandParserHelper.cpp.

### 10.44.3 Member Function Documentation

#### 10.44.3.1 void TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 179 of file DemandParserHelper.cpp.

References [TRADEMGEN::DemandParserHelper::ParserSemanticAction::\\_demand](#), and [TRADEMGEN::DemandStruct::\\_nonRefundableDisutility](#).

### 10.44.4 Member Data Documentation

#### 10.44.4.1 [DemandStruct](#)& [TRADEMGEN::DemandParserHelper::ParserSemanticAction::\\_demand](#) [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by [TRADEMGEN::DemandParserHelper::doEndDemand::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeDTD::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeTimeValue::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeWTP::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storePrefDepTime::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeFFProbMass::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeFFCode::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeStayProbMass::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeStayCode::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeTripProbMass::operator\(\)](#), and [TRADEMGEN::DemandParserHelper::storeTripCode::operator\(\)](#).

TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

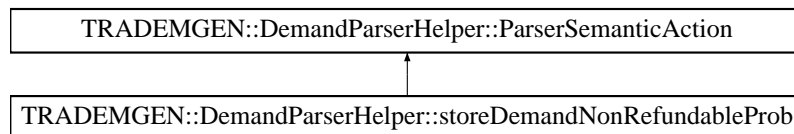
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.45 TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::



### Public Member Functions

- [storeDemandNonRefundableProb](#) ([DemandStruct](#) &)
- void [operator\(\)](#) (double iReal) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

#### 10.45.1 Detailed Description

Store the demand non refundable probability.

Definition at line 119 of file DemandParserHelper.hpp.

#### 10.45.2 Constructor & Destructor Documentation

##### 10.45.2.1 TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::storeDemandNonRefundableProb ([DemandStruct](#) &)

Actor Constructor.

Definition at line 162 of file DemandParserHelper.cpp.

### 10.45.3 Member Function Documentation

#### 10.45.3.1 void TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 167 of file DemandParserHelper.cpp.

References TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand, and TRADEMGEN::DemandStruct::\_nonRefundableProb.

### 10.45.4 Member Data Documentation

#### 10.45.4.1 DemandStruct& TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(), TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGEN::DemandParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(), TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(), TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

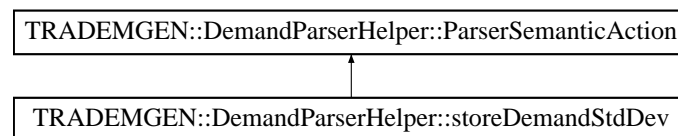
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.46 TRADEMGEN::DemandParserHelper::storeDemandStdDev Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeDemandStdDev::



### Public Member Functions

- [storeDemandStdDev](#) ([DemandStruct](#) &)
- [void operator\(\)](#) (double iReal) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

#### 10.46.1 Detailed Description

Store the demand standard deviation value.

Definition at line 95 of file DemandParserHelper.hpp.

#### 10.46.2 Constructor & Destructor Documentation

##### 10.46.2.1 TRADEMGEN::DemandParserHelper::storeDemandStdDev::storeDemandStdDev ([DemandStruct](#) &)

Actor Constructor.

Definition at line 128 of file DemandParserHelper.cpp.

#### 10.46.3 Member Function Documentation

##### 10.46.3.1 void TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator() (double *i-Real*) const

Actor Function (functor).

Definition at line 133 of file DemandParserHelper.cpp.

References [TRADEMGEN::DemandParserHelper::ParserSemanticAction::\\_demand](#), and [TRADEMGEN::DemandStruct::\\_demandStdDev](#).

#### 10.46.4 Member Data Documentation

##### 10.46.4.1 [DemandStruct](#)& [TRADEMGEN::DemandParserHelper::ParserSemanticAction::\\_demand](#) [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(), TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGEN::DemandParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(), TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(), TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

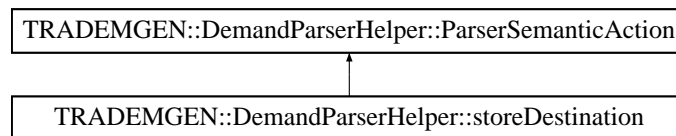
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.47 TRADEMGEN::DemandParserHelper::storeDestination Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeDestination::



### Public Member Functions

- [storeDestination](#) ([DemandStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

### 10.47.1 Detailed Description

Store the destination.

Definition at line 71 of file DemandParserHelper.hpp.

### 10.47.2 Constructor & Destructor Documentation

#### 10.47.2.1 TRADEMGEN::DemandParserHelper::storeDestination::storeDestination (DemandStruct &)

Actor Constructor.

Definition at line 92 of file DemandParserHelper.cpp.

### 10.47.3 Member Function Documentation

#### 10.47.3.1 void TRADEMGEN::DemandParserHelper::storeDestination::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 97 of file DemandParserHelper.cpp.

References TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand, and TRADEMGEN::DemandStruct::\_destination.

### 10.47.4 Member Data Documentation

#### 10.47.4.1 DemandStruct& TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(), TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGEN::DemandParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(), TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(), TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), operator(),

TRADEMGENT::DemandParserHelper::storeOrigin::operator>(), TRADEMGENT::DemandParserHelper::storeDow::operator>(), TRADEMGENT::DemandParserHelper::storePrefDepDateRangeEnd::operator>(), and TRADEMGENT::DemandParserHelper::storePrefDepDateRangeStart::operator()).

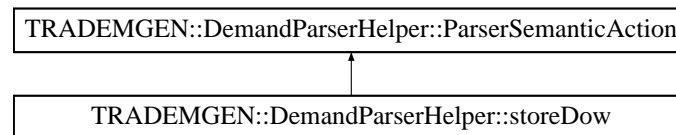
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.48 TRADEMGENT::DemandParserHelper::storeDow Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGENT::DemandParserHelper::storeDow::



### Public Member Functions

- [storeDow](#) ([DemandStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

### 10.48.1 Detailed Description

Store the DOW (day of the Week).

Definition at line 55 of file DemandParserHelper.hpp.

### 10.48.2 Constructor & Destructor Documentation

#### 10.48.2.1 TRADEMGENT::DemandParserHelper::storeDow::storeDow ([DemandStruct](#) &)

Actor Constructor.

Definition at line 70 of file DemandParserHelper.cpp.

### 10.48.3 Member Function Documentation

#### 10.48.3.1 void TRADEMGENT::DemandParserHelper::storeDow::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 75 of file DemandParserHelper.cpp.



References TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand, and  
TRADEMGENT::DemandStruct::\_dow.

#### 10.48.4 Member Data Documentation

##### 10.48.4.1 DemandStruct& TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(),  
TRADEMGENT::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGENT::Demand-  
ParserHelper::storeDTD::operator(), TRADEMGENT::DemandParserHelper::storeTimeValue-  
ProbMass::operator(), TRADEMGENT::DemandParserHelper::storeTimeValue::operator(),  
TRADEMGENT::DemandParserHelper::storeWTP::operator(), TRADEMGENT::DemandParser-  
Helper::storePrefDepTimeProbMass::operator(), TRADEMGENT::DemandParserHelper::store-  
PrefDepTime::operator(), TRADEMGENT::DemandParserHelper::storeFFProbMass::operator(),  
TRADEMGENT::DemandParserHelper::storeFFCode::operator(), TRADEMGENT::Demand-  
ParserHelper::storeStayProbMass::operator(), TRADEMGENT::DemandParserHelper::store-  
StayCode::operator(), TRADEMGENT::DemandParserHelper::storeTripProbMass::operator(),  
TRADEMGENT::DemandParserHelper::storeTripCode::operator(), TRADEMGENT::Demand-  
ParserHelper::storeChannelProbMass::operator(), TRADEMGENT::DemandParserHelper::store-  
ChannelCode::operator(), TRADEMGENT::DemandParserHelper::storePosProbMass::operator(),  
TRADEMGENT::DemandParserHelper::storePosCode::operator(), TRADEMGENT::Demand-  
ParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGENT::Demand-  
ParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGENT::DemandParser-  
Helper::storeDemandChangeFeeDisutility::operator(), TRADEMGENT::DemandParserHelper::store-  
DemandChangeFeeProb::operator(), TRADEMGENT::DemandParserHelper::storeDemand-  
StdDev::operator(), TRADEMGENT::DemandParserHelper::storeDemandMean::operator(),  
TRADEMGENT::DemandParserHelper::storePrefCabin::operator(), TRADEMGENT::DemandParser-  
Helper::storeDestination::operator(), TRADEMGENT::DemandParserHelper::storeOrigin::operator(),  
operator(), TRADEMGENT::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and  
TRADEMGENT::DemandParserHelper::storePrefDepDateRangeStart::operator().

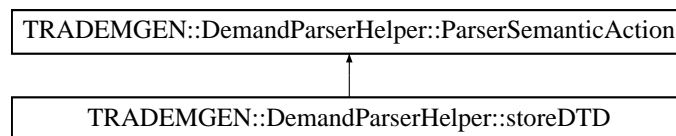
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

#### 10.49 TRADEMGEN::DemandParserHelper::storeDTD Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGENT::DemandParserHelper::storeDTD::



**Public Member Functions**

- [storeDTD](#) ([DemandStruct](#) &)
- void [operator\(\)](#) (unsigned int iInteger) const

**Public Attributes**

- [DemandStruct](#) & [\\_demand](#)

**10.49.1 Detailed Description**

Store the parameters for the arrival pattern (as expressed in DTD) continuous probability distribution.

Definition at line 258 of file DemandParserHelper.hpp.

**10.49.2 Constructor & Destructor Documentation****10.49.2.1 TRADEMGEN::DemandParserHelper::storeDTD::storeDTD ([DemandStruct](#) &)**

Actor Constructor.

Definition at line 428 of file DemandParserHelper.cpp.

**10.49.3 Member Function Documentation****10.49.3.1 void TRADEMGEN::DemandParserHelper::storeDTD::operator() (unsigned int *iInteger*) const**

Actor Function (functor).

Definition at line 433 of file DemandParserHelper.cpp.

References [TRADEMGEN::DemandParserHelper::ParserSemanticAction::\\_demand](#), and [TRADEMGEN::DemandStruct::\\_itDTD](#).

**10.49.4 Member Data Documentation****10.49.4.1 [DemandStruct](#)& [TRADEMGEN::DemandParserHelper::ParserSemanticAction::\\_demand](#) [inherited]**

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by [TRADEMGEN::DemandParserHelper::doEndDemand::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeDTProbMass::operator\(\)](#), [operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeTimeValue::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeWTP::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storePrefDepTime::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeFFProbMass::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeFFCode::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeStayProbMass::operator\(\)](#), and [TRADEMGEN::DemandParserHelper::storeStayCode::operator\(\)](#).

TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

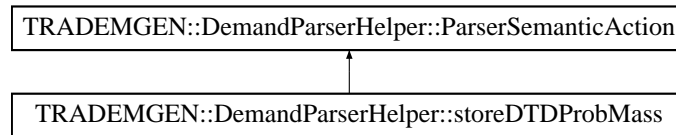
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.50 TRADEMGEN::DemandParserHelper::storeDTDProbMass Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeDTDProbMass::



### Public Member Functions

- [storeDTDProbMass](#) ([DemandStruct](#) &)
- [operator\(\)](#) (double iReal) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

#### 10.50.1 Detailed Description

Store the parameters for the arrival pattern (as expressed in DTD) continuous probability distribution.

Definition at line 267 of file DemandParserHelper.hpp.

## 10.50.2 Constructor & Destructor Documentation

### 10.50.2.1 TRADEMGEN::DemandParserHelper::storeDTDProbMass::storeDTDProbMass (DemandStruct &)

Actor Constructor.

Definition at line 440 of file DemandParserHelper.cpp.

## 10.50.3 Member Function Documentation

### 10.50.3.1 void TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator() (double *i-Real*) const

Actor Function (functor).

Definition at line 445 of file DemandParserHelper.cpp.

References TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand, TRADEMGEN::DemandStruct::\_dtdProbDist, and TRADEMGEN::DemandStruct::\_itDTD.

## 10.50.4 Member Data Documentation

### 10.50.4.1 DemandStruct& TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(), operator(), TRADEMGEN::DemandParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(), TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(), TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

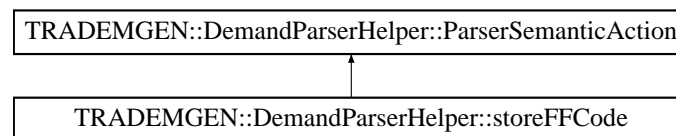
The documentation for this struct was generated from the following files:

- [trademgen/command/DemandParserHelper.hpp](#)
- [trademgen/command/DemandParserHelper.cpp](#)

## 10.51 TRADEMGEN::DemandParserHelper::storeFFCode Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeFFCode::



### Public Member Functions

- [storeFFCode](#) ([DemandStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

#### 10.51.1 Detailed Description

Store the frequent flyer code.

Definition at line 199 of file DemandParserHelper.hpp.

#### 10.51.2 Constructor & Destructor Documentation

##### 10.51.2.1 TRADEMGEN::DemandParserHelper::storeFFCode::storeFFCode ([DemandStruct](#) &)

Actor Constructor.

Definition at line 323 of file DemandParserHelper.cpp.

#### 10.51.3 Member Function Documentation

##### 10.51.3.1 void TRADEMGEN::DemandParserHelper::storeFFCode::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 328 of file DemandParserHelper.cpp.

References [TRADEMGEN::DemandParserHelper::ParserSemanticAction::\\_demand](#), and [TRADEMGEN::DemandStruct::\\_itFFCode](#).

### 10.51.4 Member Data Documentation

#### 10.51.4.1 DemandStruct& TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(), TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGEN::DemandParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(), TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

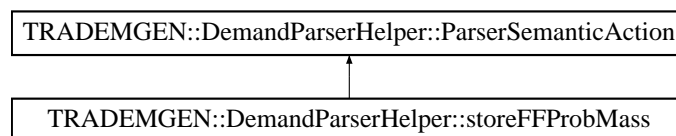
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.52 TRADEMGEN::DemandParserHelper::storeFFProbMass Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeFFProbMass::



**Public Member Functions**

- [storeFFProbMass](#) ([DemandStruct](#) &)
- void [operator\(\)](#) (double iReal) const

**Public Attributes**

- [DemandStruct](#) & [\\_demand](#)

**10.52.1 Detailed Description**

Store the frequent flyer probability mass.

Definition at line 207 of file DemandParserHelper.hpp.

**10.52.2 Constructor & Destructor Documentation****10.52.2.1 TRADEMGEN::DemandParserHelper::storeFFProbMass::storeFFProbMass ([DemandStruct](#) &)**

Actor Constructor.

Definition at line 334 of file DemandParserHelper.cpp.

**10.52.3 Member Function Documentation****10.52.3.1 void TRADEMGEN::DemandParserHelper::storeFFProbMass::operator() (double *i-Real*) const**

Actor Function (functor).

Definition at line 339 of file DemandParserHelper.cpp.

References [TRADEMGEN::DemandParserHelper::ParserSemanticAction::\\_demand](#), [TRADEMGEN::DemandStruct::\\_ffProbDist](#), and [TRADEMGEN::DemandStruct::\\_itFFCode](#).

**10.52.4 Member Data Documentation****10.52.4.1 [DemandStruct&](#) [TRADEMGEN::DemandParserHelper::ParserSemanticAction::\\_demand](#) [inherited]**

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by [TRADEMGEN::DemandParserHelper::doEndDemand::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeDTD::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeTimeValue::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeWTP::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storePrefDepTime::operator\(\)](#), [operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeFFCode::operator\(\)](#), [TRADEMGEN::DemandParserHelper::storeStayProbMass::operator\(\)](#), and [TRADEMGEN::DemandParserHelper::storeStayCode::operator\(\)](#).

TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

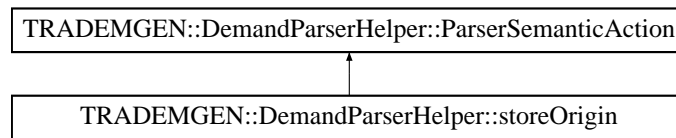
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.53 TRADEMGEN::DemandParserHelper::storeOrigin Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeOrigin::



### Public Member Functions

- [storeOrigin](#) ([DemandStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

#### 10.53.1 Detailed Description

Store the origin.

Definition at line 63 of file DemandParserHelper.hpp.

#### 10.53.2 Constructor & Destructor Documentation

##### 10.53.2.1 TRADEMGEN::DemandParserHelper::storeOrigin::storeOrigin ([DemandStruct](#) &)



Actor Constructor.

Definition at line 81 of file DemandParserHelper.cpp.

### 10.53.3 Member Function Documentation

#### 10.53.3.1 void TRADEMGEN::DemandParserHelper::storeOrigin::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 86 of file DemandParserHelper.cpp.

References TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand, and TRADEMGEN::DemandStruct::\_origin.

### 10.53.4 Member Data Documentation

#### 10.53.4.1 DemandStruct& TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(), TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGEN::DemandParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(), TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(), TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::storeDestination::operator(), operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

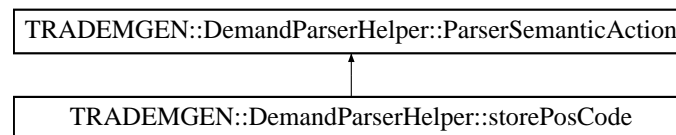
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.54 TRADEMGEN::DemandParserHelper::storePosCode Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storePosCode::



### Public Member Functions

- [storePosCode](#) ([DemandStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

#### 10.54.1 Detailed Description

Store the pos type code.

Definition at line 135 of file DemandParserHelper.hpp.

#### 10.54.2 Constructor & Destructor Documentation

##### 10.54.2.1 TRADEMGEN::DemandParserHelper::storePosCode::storePosCode ([DemandStruct](#) &)

Actor Constructor.

Definition at line 185 of file DemandParserHelper.cpp.

#### 10.54.3 Member Function Documentation

##### 10.54.3.1 void TRADEMGEN::DemandParserHelper::storePosCode::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 190 of file DemandParserHelper.cpp.

References [TRADEMG...::DemandParserHelper::ParserSemanticAction::\\_demand](#), and [TRADEMG...::DemandStruct::\\_itPosCode](#).

#### 10.54.4 Member Data Documentation

##### 10.54.4.1 [DemandStruct](#)& [TRADEMG...::DemandParserHelper::ParserSemanticAction::\\_demand](#) [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(),  
 TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGEN::Demand-  
 ParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue-  
 ProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(),  
 TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::Demand-  
 ParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParser-  
 Helper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::store-  
 FFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(),  
 TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::Demand-  
 ParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTrip-  
 ProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(),  
 TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::Demand-  
 ParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::store-  
 PosProbMass::operator(), operator(), TRADEMGEN::DemandParserHelper::storeDemand-  
 NonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemand-  
 NonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChange-  
 FeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFee-  
 Prob::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(),  
 TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::Demand-  
 ParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::store-  
 Destination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(),  
 TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParser-  
 Helper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::store-  
 PrefDepDateRangeStart::operator().

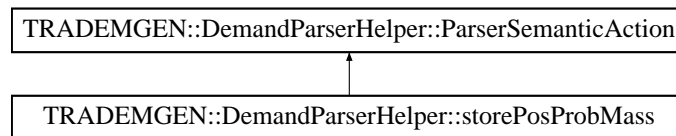
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.55 TRADEMGEN::DemandParserHelper::storePosProbMass Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storePosProbMass:



### Public Member Functions

- [storePosProbMass](#) ([DemandStruct](#) &)
- void [operator\(\)](#) (double iReal) const

**Public Attributes**

- [DemandStruct](#) & [\\_demand](#)

**10.55.1 Detailed Description**

Store the pos type probability mass.

Definition at line 143 of file DemandParserHelper.hpp.

**10.55.2 Constructor & Destructor Documentation****10.55.2.1 TRADEMGEN::DemandParserHelper::storePosProbMass::storePosProbMass ([DemandStruct](#) &)**

Actor Constructor.

Definition at line 197 of file DemandParserHelper.cpp.

**10.55.3 Member Function Documentation****10.55.3.1 void TRADEMGEN::DemandParserHelper::storePosProbMass::operator() (double *i-Real*) const**

Actor Function (functor).

Definition at line 202 of file DemandParserHelper.cpp.

References [TRADEMGENT::DemandParserHelper::ParserSemanticAction::\\_demand](#), [TRADEMGENT::DemandStruct::\\_itPosCode](#), and [TRADEMGENT::DemandStruct::\\_posProbDist](#).

**10.55.4 Member Data Documentation****10.55.4.1 [DemandStruct](#)& [TRADEMGENT::DemandParserHelper::ParserSemanticAction::\\_demand](#) [inherited]**

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by [TRADEMGENT::DemandParserHelper::doEndDemand::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeDTDProbMass::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeDTD::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeTimeValueProbMass::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeTimeValue::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeWTP::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storePrefDepTimeProbMass::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storePrefDepTime::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeFFProbMass::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeFFCode::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeStayProbMass::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeStayCode::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeTripProbMass::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeTripCode::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeChannelProbMass::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeChannelCode::operator\(\)](#), [operator\(\)](#), [TRADEMGENT::DemandParserHelper::storePosCode::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeDemandNonRefundableDisutility::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeDemandNonRefundableProb::operator\(\)](#), and [TRADEMGENT::DemandParserHelper::storeDemandChange](#).

FeeDisutility::operator>(), TRADEMGEN::DemandParserHelper::storeDemandChangeFee-  
 Prob::operator>(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator>(),  
 TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::Demand-  
 ParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::store-  
 Destination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(),  
 TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParser-  
 Helper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::store-  
 PrefDepDateRangeStart::operator()).

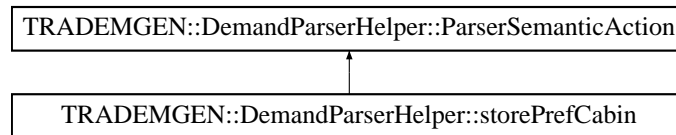
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.56 TRADEMGEN::DemandParserHelper::storePrefCabin Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storePrefCabin::



### Public Member Functions

- [storePrefCabin](#) ([DemandStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

#### 10.56.1 Detailed Description

Store the preferred cabin.

Definition at line 79 of file DemandParserHelper.hpp.

#### 10.56.2 Constructor & Destructor Documentation

##### 10.56.2.1 TRADEMGEN::DemandParserHelper::storePrefCabin::storePrefCabin ([DemandStruct](#) &)

Actor Constructor.

Definition at line 104 of file DemandParserHelper.cpp.

### 10.56.3 Member Function Documentation

#### 10.56.3.1 void TRADEMGEN::DemandParserHelper::storePrefCabin::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 109 of file DemandParserHelper.cpp.

References TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand, and TRADEMGEN::DemandStruct::\_prefCabin.

### 10.56.4 Member Data Documentation

#### 10.56.4.1 DemandStruct& TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator>(), TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGEN::DemandParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(), TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(), TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), operator(), TRADEMGEN::DemandParserHelper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

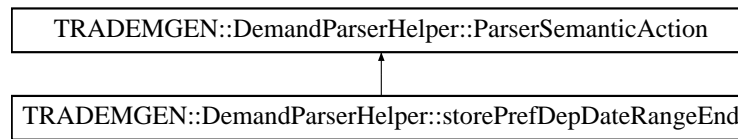
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.57 TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::



## Public Member Functions

- [storePrefDepDateRangeEnd](#) ([DemandStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

## Public Attributes

- [DemandStruct](#) & [\\_demand](#)

### 10.57.1 Detailed Description

Store the end of the date range.

Definition at line 47 of file DemandParserHelper.hpp.

### 10.57.2 Constructor & Destructor Documentation

#### 10.57.2.1 TRADEMGEM::DemandParserHelper::storePrefDepDateRangeEnd::storePrefDepDateRangeEnd ([DemandStruct](#) &)

Actor Constructor.

Definition at line 47 of file DemandParserHelper.cpp.

### 10.57.3 Member Function Documentation

#### 10.57.3.1 void TRADEMGEM::DemandParserHelper::storePrefDepDateRangeEnd::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 52 of file DemandParserHelper.cpp.

References [TRADEMGEM::DemandStruct::\\_dateRange](#), [TRADEMGEM::DemandParserHelper::ParserSemanticAction::\\_demand](#), [TRADEMGEM::DemandStruct::\\_itSeconds](#), [TRADEMGEM::DemandStruct::\\_prefDepDateEnd](#), [TRADEMGEM::DemandStruct::\\_prefDepDateStart](#), and [TRADEMGEM::DemandStruct::getDate\(\)](#).

### 10.57.4 Member Data Documentation

#### 10.57.4.1 [DemandStruct](#)& [TRADEMGEM::DemandParserHelper::ParserSemanticAction::\\_demand](#) [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(),  
 TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGEN::Demand-  
 ParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue-  
 ProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(),  
 TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParser-  
 Helper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::store-  
 PrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(),  
 TRADEMGEN::DemandParserHelper::storeFFCode::operator(), TRADEMGEN::Demand-  
 ParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::store-  
 StayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(),  
 TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::Demand-  
 ParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::store-  
 ChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(),  
 TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::Demand-  
 ParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::Demand-  
 ParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParser-  
 Helper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::store-  
 DemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemand-  
 StdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(),  
 TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::Demand-  
 ParserHelper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::store-  
 Origin::operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), operator(), and  
 TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

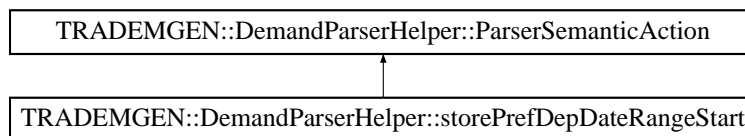
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.58 TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::



### Public Member Functions

- [storePrefDepDateRangeStart](#) ([DemandStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)



### 10.58.1 Detailed Description

Store the start of the date range.

Definition at line 39 of file DemandParserHelper.hpp.

### 10.58.2 Constructor & Destructor Documentation

#### 10.58.2.1 TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::storePrefDepDateRangeStart (DemandStruct &)

Actor Constructor.

Definition at line 32 of file DemandParserHelper.cpp.

### 10.58.3 Member Function Documentation

#### 10.58.3.1 void TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator()(iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 37 of file DemandParserHelper.cpp.

References TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand, TRADEMGEN::DemandStruct::\_itSeconds, TRADEMGEN::DemandStruct::\_prefDepDateStart, and TRADEMGEN::DemandStruct::getDate().

### 10.58.4 Member Data Documentation

#### 10.58.4.1 DemandStruct& TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(), TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGEN::DemandParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(), TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(), TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), and TRADEMGEN::DemandParserHelper::storeDemandMean::operator().

TRADEMGEN::DemandParserHelper::storePrefCabin::operator>(), TRADEMGEN::DemandParserHelper::storeDestination::operator>(), TRADEMGEN::DemandParserHelper::storeOrigin::operator>(), TRADEMGEN::DemandParserHelper::storeDow::operator>(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator>(), and operator().

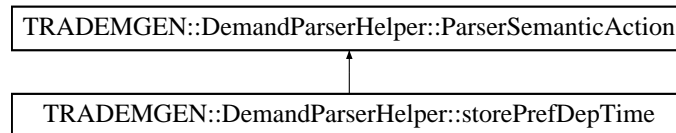
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.59 TRADEMGEN::DemandParserHelper::storePrefDepTime Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storePrefDepTime::



### Public Member Functions

- [storePrefDepTime](#) ([DemandStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

#### 10.59.1 Detailed Description

Store the parameters for the preferred departure time continuous probability distribution.

Definition at line 216 of file DemandParserHelper.hpp.

#### 10.59.2 Constructor & Destructor Documentation

##### 10.59.2.1 TRADEMGEN::DemandParserHelper::storePrefDepTime::storePrefDepTime ([DemandStruct](#) &)

Actor Constructor.

Definition at line 357 of file DemandParserHelper.cpp.

#### 10.59.3 Member Function Documentation

##### 10.59.3.1 void TRADEMGEN::DemandParserHelper::storePrefDepTime::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 362 of file DemandParserHelper.cpp.

References TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand, TRADEMGEN::DemandStruct::\_itMinutes, TRADEMGEN::DemandStruct::\_itPrefDepTime, TRADEMGEN::DemandStruct::\_itSeconds, and TRADEMGEN::DemandStruct::getTime().

## 10.59.4 Member Data Documentation

### 10.59.4.1 DemandStruct & TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(), TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGEN::DemandParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(), TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(), TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

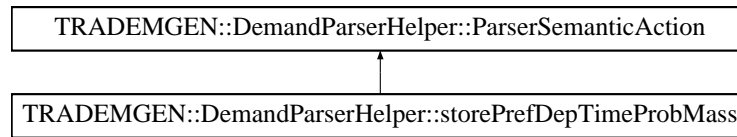
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.60 TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::



## Public Member Functions

- [storePrefDepTimeProbMass](#) ([DemandStruct](#) &)
- [void operator\(\)](#) (double iReal) const

## Public Attributes

- [DemandStruct](#) & [\\_demand](#)

### 10.60.1 Detailed Description

Store the parameters for the preferred departure time continuous probability distribution.

Definition at line 225 of file DemandParserHelper.hpp.

### 10.60.2 Constructor & Destructor Documentation

#### 10.60.2.1 TRADEMGEM::DemandParserHelper::storePrefDepTimeProbMass::storePrefDepTimeProbMass ([DemandStruct](#) &)

Actor Constructor.

Definition at line 377 of file DemandParserHelper.cpp.

### 10.60.3 Member Function Documentation

#### 10.60.3.1 void TRADEMGEM::DemandParserHelper::storePrefDepTimeProbMass::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 382 of file DemandParserHelper.cpp.

References [TRADEMGEM::DemandParserHelper::ParserSemanticAction::\\_demand](#), [TRADEMGEM::DemandStruct::\\_itPrefDepTime](#), and [TRADEMGEM::DemandStruct::\\_prefDepTimeProbDist](#).

### 10.60.4 Member Data Documentation

#### 10.60.4.1 [DemandStruct](#)& [TRADEMGEM::DemandParserHelper::ParserSemanticAction::\\_demand](#) [[inherited](#)]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(),  
 TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGEN::Demand-  
 ParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::store-  
 TimeValueProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTime-  
 Value::operator(), TRADEMGEN::DemandParserHelper::storeWTP::operator(), operator(),  
 TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::Demand-  
 ParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::store-  
 FFCode::operator(), TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(),  
 TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::Demand-  
 ParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTrip-  
 Code::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(),  
 TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::Demand-  
 ParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::store-  
 PosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundable-  
 Disutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundable-  
 Prob::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFee-  
 Disutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFee-  
 Prob::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(),  
 TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::Demand-  
 ParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::store-  
 Destination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(),  
 TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParser-  
 Helper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::store-  
 PrefDepDateRangeStart::operator().

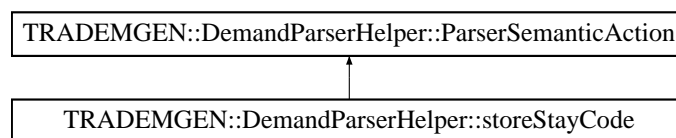
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.61 TRADEMGEN::DemandParserHelper::storeStayCode Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeStayCode::



### Public Member Functions

- [storeStayCode](#) ([DemandStruct](#) &)
- void [operator\(\)](#) (unsigned int iInteger) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

### 10.61.1 Detailed Description

Store the stay type code.

Definition at line 183 of file DemandParserHelper.hpp.

### 10.61.2 Constructor & Destructor Documentation

#### 10.61.2.1 TRADEMGEN::DemandParserHelper::storeStayCode::storeStayCode ([DemandStruct](#) &)

Actor Constructor.

Definition at line 289 of file DemandParserHelper.cpp.

### 10.61.3 Member Function Documentation

#### 10.61.3.1 void TRADEMGEN::DemandParserHelper::storeStayCode::operator() (unsigned int *i-Integer*) const

Actor Function (functor).

Definition at line 294 of file DemandParserHelper.cpp.

References [TRADEMGENT::DemandParserHelper::ParserSemanticAction::\\_demand](#), and [TRADEMGENT::DemandStruct::\\_itStayDuration](#).

### 10.61.4 Member Data Documentation

#### 10.61.4.1 [DemandStruct&](#) [TRADEMGENT::DemandParserHelper::ParserSemanticAction::\\_demand](#) [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by [TRADEMGENT::DemandParserHelper::doEndDemand::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeDTDProbMass::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeDTD::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeTimeValueProbMass::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeTimeValue::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeWTP::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storePrefDepTimeProbMass::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storePrefDepTime::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeFFProbMass::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeFFCode::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeStayProbMass::operator\(\)](#), [operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeTripProbMass::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeTripCode::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeChannelProbMass::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeChannelCode::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storePosProbMass::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storePosCode::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeDemandNonRefundableDisutility::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeDemandNonRefundableProb::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeDemandChangeFeeDisutility::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeDemandChangeFeeProb::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeDemandStdDev::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storeDemandMean::operator\(\)](#), [TRADEMGENT::DemandParserHelper::storePrefCabin::operator\(\)](#), [TRADEMGENT::DemandParserHelper::store-](#)

Destination::operator()), TRADEMGEN::DemandParserHelper::storeOrigin::operator()),  
 TRADEMGEN::DemandParserHelper::storeDow::operator()), TRADEMGEN::DemandParser-  
 Helper::storePrefDepDateRangeEnd::operator()), and TRADEMGEN::DemandParserHelper::store-  
 PrefDepDateRangeStart::operator()).

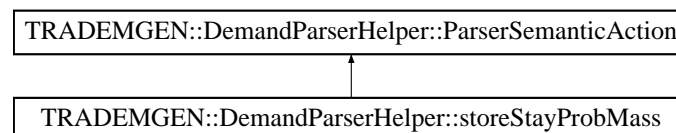
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.62 TRADEMGEN::DemandParserHelper::storeStayProbMass Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeStayProbMass::



### Public Member Functions

- [storeStayProbMass](#) ([DemandStruct](#) &)
- void [operator\(\)](#) (double iReal) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

#### 10.62.1 Detailed Description

Store the stay type probability mass.

Definition at line 191 of file DemandParserHelper.hpp.

#### 10.62.2 Constructor & Destructor Documentation

##### 10.62.2.1 TRADEMGEN::DemandParserHelper::storeStayProbMass::storeStayProbMass ([DemandStruct](#) &)

Actor Constructor.

Definition at line 301 of file DemandParserHelper.cpp.

#### 10.62.3 Member Function Documentation

##### 10.62.3.1 void TRADEMGEN::DemandParserHelper::storeStayProbMass::operator() (double *i-Real*) const

Actor Function (functor).

Definition at line 306 of file DemandParserHelper.cpp.

References TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand, TRADEMGEN::DemandStruct::\_itStayDuration, and TRADEMGEN::DemandStruct::\_stayProbDist.

## 10.62.4 Member Data Documentation

### 10.62.4.1 DemandStruct& TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(), TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGEN::DemandParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(), TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(), operator(), TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

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

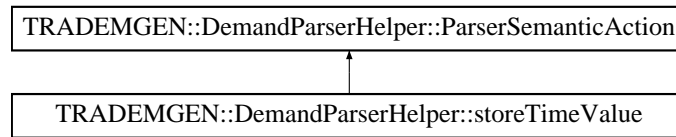
- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.63 TRADEMGEN::DemandParserHelper::storeTimeValue Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeTimeValue::





### Public Member Functions

- [storeTimeValue](#) ([DemandStruct](#) &)
- [void operator\(\)](#) (double *iReal*) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

#### 10.63.1 Detailed Description

Store the time value.

Definition at line 241 of file DemandParserHelper.hpp.

#### 10.63.2 Constructor & Destructor Documentation

##### 10.63.2.1 TRADEMGEN::DemandParserHelper::storeTimeValue::storeTimeValue ([DemandStruct](#) &)

Actor Constructor.

Definition at line 404 of file DemandParserHelper.cpp.

#### 10.63.3 Member Function Documentation

##### 10.63.3.1 void TRADEMGEN::DemandParserHelper::storeTimeValue::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 409 of file DemandParserHelper.cpp.

References [TRADEMGEM::DemandParserHelper::ParserSemanticAction::\\_demand](#), and [TRADEMGEM::DemandStruct::\\_itTimeValue](#).

#### 10.63.4 Member Data Documentation

##### 10.63.4.1 [DemandStruct](#)& [TRADEMGEM::DemandParserHelper::ParserSemanticAction::\\_demand](#) [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by [TRADEMGEM::DemandParserHelper::doEndDemand::operator\(\)](#), [TRADEMGEM::DemandParserHelper::storeDTDProbMass::operator\(\)](#), [TRADEMGEM::DemandParserHelper::storeDTD::operator\(\)](#), and [TRADEMGEM::DemandParserHelper::storeTime-](#)

ValueProbMass::operator(), operator(), TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(), TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

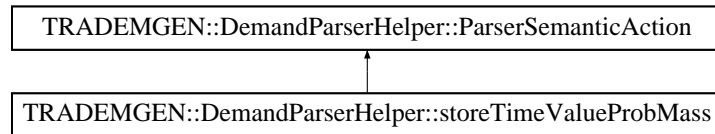
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.64 TRADEMGEN::DemandParserHelper::storeTimeValueProbMass Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::



### Public Member Functions

- [storeTimeValueProbMass](#) ([DemandStruct](#) &)
- [operator\(\)](#) (double iReal) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

### 10.64.1 Detailed Description

Store the time value probability mass.

Definition at line 249 of file DemandParserHelper.hpp.

### 10.64.2 Constructor & Destructor Documentation

#### 10.64.2.1 TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::storeTimeValueProbMass (DemandStruct &)

Actor Constructor.

Definition at line 415 of file DemandParserHelper.cpp.

### 10.64.3 Member Function Documentation

#### 10.64.3.1 void TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator()(double *iReal*) const

Actor Function (functor).

Definition at line 420 of file DemandParserHelper.cpp.

References TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand, TRADEMGEN::DemandStruct::\_itTimeValue, and TRADEMGEN::DemandStruct::\_timeValueProbDist.

### 10.64.4 Member Data Documentation

#### 10.64.4.1 DemandStruct& TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(), TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGEN::DemandParserHelper::storeDTD::operator(), operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(), TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(), TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::Demand

ParserHelper::storePrefCabin::operator>(), TRADEMGEN::DemandParserHelper::store-Destination::operator>(), TRADEMGEN::DemandParserHelper::storeOrigin::operator>(), TRADEMGEN::DemandParserHelper::storeDow::operator>(), TRADEMGEN::DemandParser-Helper::storePrefDepDateRangeEnd::operator>(), and TRADEMGEN::DemandParserHelper::store-PrefDepDateRangeStart::operator()).

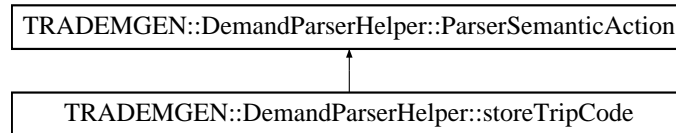
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.65 TRADEMGEN::DemandParserHelper::storeTripCode Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeTripCode::



### Public Member Functions

- [storeTripCode](#) ([DemandStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

#### 10.65.1 Detailed Description

Store the trip type code.

Definition at line 167 of file DemandParserHelper.hpp.

#### 10.65.2 Constructor & Destructor Documentation

##### 10.65.2.1 TRADEMGEN::DemandParserHelper::storeTripCode::storeTripCode ([DemandStruct](#) &)

Actor Constructor.

Definition at line 254 of file DemandParserHelper.cpp.

#### 10.65.3 Member Function Documentation

##### 10.65.3.1 void TRADEMGEN::DemandParserHelper::storeTripCode::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 259 of file DemandParserHelper.cpp.

References TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand, and TRADEMGEN::DemandStruct::\_itTripCode.

## 10.65.4 Member Data Documentation

### 10.65.4.1 DemandStruct& TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator>(), TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGEN::DemandParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(), TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(), TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

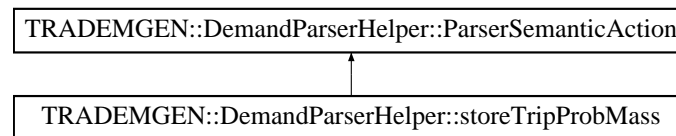
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.66 TRADEMGEN::DemandParserHelper::storeTripProbMass Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeTripProbMass::



## Public Member Functions

- [storeTripProbMass](#) ([DemandStruct](#) &)
- [void operator\(\)](#) (double iReal) const

## Public Attributes

- [DemandStruct](#) & [\\_demand](#)

### 10.66.1 Detailed Description

Store the trip type probability mass.

Definition at line 175 of file DemandParserHelper.hpp.

### 10.66.2 Constructor & Destructor Documentation

#### 10.66.2.1 TRADEMGEN::DemandParserHelper::storeTripProbMass::storeTripProbMass ([DemandStruct](#) &)

Actor Constructor.

Definition at line 266 of file DemandParserHelper.cpp.

### 10.66.3 Member Function Documentation

#### 10.66.3.1 void TRADEMGEN::DemandParserHelper::storeTripProbMass::operator() (double *i-Real*) const

Actor Function (functor).

Definition at line 271 of file DemandParserHelper.cpp.

References [TRADEMGEM::DemandParserHelper::ParserSemanticAction::\\_demand](#), [TRADEMGEM::DemandStruct::\\_itTripCode](#), and [TRADEMGEM::DemandStruct::\\_tripProbDist](#).

### 10.66.4 Member Data Documentation

#### 10.66.4.1 [DemandStruct](#)& [TRADEMGEM::DemandParserHelper::ParserSemanticAction::\\_demand](#) [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by [TRADEMGEM::DemandParserHelper::doEndDemand::operator\(\)](#), [TRADEMGEM::DemandParserHelper::storeDTDProbMass::operator\(\)](#), [TRADEMGEM::DemandParserHelper::storeDTD::operator\(\)](#), and [TRADEMGEM::DemandParserHelper::storeTimeValue-](#)

ProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(),  
 TRADEMGEN::DemandParserHelper::storeWTP::operator(), TRADEMGEN::Demand-  
 ParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParser-  
 Helper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::store-  
 FFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(),  
 TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::Demand-  
 ParserHelper::storeStayCode::operator(), operator(), TRADEMGEN::DemandParser-  
 Helper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProb-  
 Mass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(),  
 TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::Demand-  
 ParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemand-  
 NonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemand-  
 NonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChange-  
 FeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFee-  
 Prob::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(),  
 TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::Demand-  
 ParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::store-  
 Destination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(),  
 TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParser-  
 Helper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::store-  
 PrefDepDateRangeStart::operator().

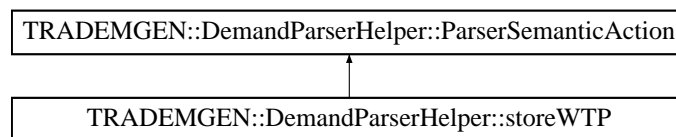
The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.67 TRADEMGEN::DemandParserHelper::storeWTP Struct Reference

```
#include <trademgen/command/DemandParserHelper.hpp>
```

Inheritance diagram for TRADEMGEN::DemandParserHelper::storeWTP::



### Public Member Functions

- [storeWTP](#) ([DemandStruct](#) &)
- void [operator\(\)](#) (double iReal) const

### Public Attributes

- [DemandStruct](#) & [\\_demand](#)

#### 10.67.1 Detailed Description

Store the parameters for the min Willingness-To-Pay (WTP).

Definition at line 233 of file DemandParserHelper.hpp.

## 10.67.2 Constructor & Destructor Documentation

### 10.67.2.1 TRADEMGEN::DemandParserHelper::storeWTP::storeWTP (DemandStruct &)

Actor Constructor.

Definition at line 393 of file DemandParserHelper.cpp.

## 10.67.3 Member Function Documentation

### 10.67.3.1 void TRADEMGEN::DemandParserHelper::storeWTP::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 398 of file DemandParserHelper.cpp.

References TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand, and TRADEMGEN::DemandStruct::\_minWTP.

## 10.67.4 Member Data Documentation

### 10.67.4.1 DemandStruct& TRADEMGEN::DemandParserHelper::ParserSemanticAction::\_demand [inherited]

Actor Context.

Definition at line 35 of file DemandParserHelper.hpp.

Referenced by TRADEMGEN::DemandParserHelper::doEndDemand::operator(), TRADEMGEN::DemandParserHelper::storeDTDProbMass::operator(), TRADEMGEN::DemandParserHelper::storeDTD::operator(), TRADEMGEN::DemandParserHelper::storeTimeValueProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTimeValue::operator(), operator(), TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass::operator(), TRADEMGEN::DemandParserHelper::storePrefDepTime::operator(), TRADEMGEN::DemandParserHelper::storeFFProbMass::operator(), TRADEMGEN::DemandParserHelper::storeFFCode::operator(), TRADEMGEN::DemandParserHelper::storeStayProbMass::operator(), TRADEMGEN::DemandParserHelper::storeStayCode::operator(), TRADEMGEN::DemandParserHelper::storeTripProbMass::operator(), TRADEMGEN::DemandParserHelper::storeTripCode::operator(), TRADEMGEN::DemandParserHelper::storeChannelProbMass::operator(), TRADEMGEN::DemandParserHelper::storeChannelCode::operator(), TRADEMGEN::DemandParserHelper::storePosProbMass::operator(), TRADEMGEN::DemandParserHelper::storePosCode::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility::operator(), TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb::operator(), TRADEMGEN::DemandParserHelper::storeDemandStdDev::operator(), TRADEMGEN::DemandParserHelper::storeDemandMean::operator(), TRADEMGEN::DemandParserHelper::storePrefCabin::operator(), TRADEMGEN::DemandParserHelper::storeDestination::operator(), TRADEMGEN::DemandParserHelper::storeOrigin::operator(), TRADEMGEN::DemandParserHelper::storeDow::operator(), TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd::operator(), and TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart::operator().

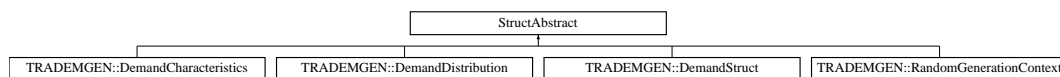


The documentation for this struct was generated from the following files:

- trademgen/command/[DemandParserHelper.hpp](#)
- trademgen/command/[DemandParserHelper.cpp](#)

## 10.68 StructAbstract Class Reference

Inheritance diagram for StructAbstract::



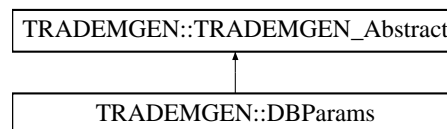
The documentation for this class was generated from the following files:

- trademgen/basic/[RandomGenerationContext.hpp](#)
- trademgen/basic/[DemandCharacteristics.hpp](#)
- trademgen/bom/[DemandStruct.hpp](#)
- trademgen/basic/[DemandDistribution.hpp](#)

## 10.69 TRADEMGEN::TRADEMGEN\_Abstract Struct Reference

```
#include <trademgen/TRADEMGEN_Abstract.hpp>
```

Inheritance diagram for TRADEMGEN::TRADEMGEN\_Abstract::



### Public Member Functions

- virtual void [toStream](#) (std::ostream &ioOut) const =0
- virtual void [fromStream](#) (std::istream &ioIn)=0
- virtual std::string [toString](#) () const =0

### Protected Member Functions

- [TRADEMGEN\\_Abstract](#) ()
- [TRADEMGEN\\_Abstract](#) (const [TRADEMGEN\\_Abstract](#) &)
- virtual [~TRADEMGEN\\_Abstract](#) ()

### 10.69.1 Detailed Description

Base class for the [TRADEMGEN](#) interface structures.

Definition at line 16 of file [TRADEMGEN\\_Abstract.hpp](#).

### 10.69.2 Constructor & Destructor Documentation

#### 10.69.2.1 TRADEMGEN::TRADEMGEN\_Abstract::TRADEMGEN\_Abstract () [inline, protected]

Protected Default Constructor to ensure this class is abstract.

Definition at line 33 of file TRADEMGEN\_Abstract.hpp.

#### 10.69.2.2 TRADEMGEN::TRADEMGEN\_Abstract::TRADEMGEN\_Abstract (const TRADEMGEN\_Abstract &) [inline, protected]

Definition at line 34 of file TRADEMGEN\_Abstract.hpp.

#### 10.69.2.3 virtual TRADEMGEN::TRADEMGEN\_Abstract::~~TRADEMGEN\_Abstract () [inline, protected, virtual]

Destructor.

Definition at line 37 of file TRADEMGEN\_Abstract.hpp.

### 10.69.3 Member Function Documentation

#### 10.69.3.1 virtual void TRADEMGEN::TRADEMGEN\_Abstract::toStream (std::ostream & ioOut) const [pure virtual]

Dump a structure into an output stream.

##### Parameters:

*ostream&* the output stream.

Implemented in [TRADEMGEN::DBParams](#).

#### 10.69.3.2 virtual void TRADEMGEN::TRADEMGEN\_Abstract::fromStream (std::istream & ioIn) [pure virtual]

Read a structure from an input stream.

##### Parameters:

*istream&* the input stream.

Implemented in [TRADEMGEN::DBParams](#).

Referenced by operator>>().

#### 10.69.3.3 virtual std::string TRADEMGEN::TRADEMGEN\_Abstract::toString () const [pure virtual]

Get the serialised version of the structure.

Implemented in [TRADEMGEN::DBParams](#).

The documentation for this struct was generated from the following file:

- trademgen/[TRADEMGEN\\_Abstract.hpp](#)

## 10.70 TRADEMGEN::TRADEMGEN\_Service Class Reference

class holding the services related to Travel Demand Generation.

```
#include <trademgen/TRADEMGEN_Service.hpp>
```

### Public Member Functions

- [TRADEMGEN\\_Service](#) (const stdair::BasLogParams &, const stdair::BasDBParams &, const stdair::RandomSeed\_T &)

*Constructor.*

- [TRADEMGEN\\_Service](#) (const stdair::BasLogParams &, const stdair::RandomSeed\_T &)
- [TRADEMGEN\\_Service](#) (stdair::STDAIR\_ServicePtr\_T, SEVMGR::SEVMGR\_ServicePtr\_T, const stdair::RandomSeed\_T &)
- void [parseAndLoad](#) (const [DemandFilePath](#) &)
- [~TRADEMGEN\\_Service](#) ()
- void [buildSampleBom](#) ()
- void [clonePersistentBom](#) ()
- void [buildComplementaryLinks](#) (stdair::BomRoot &)
- stdair::BookingRequestStruct [buildSampleBookingRequest](#) (const bool isForCRS=false)
- void [displayAirlineListFromDB](#) () const
- const stdair::Count\_T & [getExpectedTotalNumberOfRequestsToBeGenerated](#) () const
- const stdair::Count\_T & [getActualTotalNumberOfRequestsToBeGenerated](#) () const
- const bool [stillHavingRequestsToBeGenerated](#) (const stdair::DemandStreamKeyStr\_T &, stdair::ProgressStatusSet &, const stdair::DemandGenerationMethod &) const
- stdair::Count\_T [generateFirstRequests](#) (const stdair::DemandGenerationMethod &) const
- stdair::BookingRequestPtr\_T [generateNextRequest](#) (const stdair::DemandStreamKeyStr\_T &, const stdair::DemandGenerationMethod &) const
- bool [hasDemandStream](#) (const stdair::DemandStreamKeyStr\_T &) const
- stdair::ProgressStatusSet [popEvent](#) (stdair::EventStruct &) const
- bool [isQueueDone](#) () const
- bool [generateCancellation](#) (const stdair::TravelSolutionStruct &, const stdair::PartySize\_T &, const stdair::DateTime\_T &, const stdair::Date\_T &) const
- void [reset](#) () const
- const stdair::ProgressStatus & [getProgressStatus](#) () const
- const stdair::ProgressStatus & [getProgressStatus](#) (const stdair::EventType::EN\_EventType &) const
- std::string [jsonHandler](#) (const stdair::JSONString &) const
- std::string [csvDisplay](#) () const
- std::string [list](#) () const
- std::string [list](#) (const stdair::EventType::EN\_EventType &) const
- std::string [displayDemandStream](#) () const

### 10.70.1 Detailed Description

class holding the services related to Travel Demand Generation.

Definition at line 44 of file TRADEMGEN\_Service.hpp.

## 10.70.2 Constructor & Destructor Documentation

### 10.70.2.1 TRADEMGEN::TRADEMGEN\_Service::TRADEMGEN\_Service (const stdair::BasLogParams &, const stdair::BasDBParams &, const stdair::RandomSeed\_T &)

Constructor.

The `initTrademgenService()` method is called; see the corresponding documentation for more details.

A reference on an output stream is given, so that log outputs can be directed onto that stream.

Moreover, database connection parameters are given, so that a session can be created on the corresponding database.

#### Parameters:

*const* stdair::BasLogParams& Parameters for the output log stream.

*const* stdair::BasDBParams& Parameters for the database access.

*const* stdair::RandomSeed\_T& Seed for the random generation.

Definition at line 78 of file `TRADEMGEN_Service.cpp`.

### 10.70.2.2 TRADEMGEN::TRADEMGEN\_Service::TRADEMGEN\_Service (const stdair::BasLogParams &, const stdair::RandomSeed\_T &)

Constructor.

The `initTrademgenService()` method is called; see the corresponding documentation for more details.

A reference on an output stream is given, so that log outputs can be directed onto that stream.

#### Parameters:

*const* stdair::BasLogParams& Parameters for the output log stream.

*const* stdair::RandomSeed\_T& Seed for the random generation.

Definition at line 54 of file `TRADEMGEN_Service.cpp`.

### 10.70.2.3 TRADEMGEN::TRADEMGEN\_Service::TRADEMGEN\_Service (stdair::STDAIR\_ServicePtr\_T, SEVMGR::SEVMGR\_ServicePtr\_T, const stdair::RandomSeed\_T &)

Constructor.

The `initTrademgenService()` method is called; see the corresponding documentation for more details.

Moreover, as no reference on any output stream is given, neither any database access parameter is given, it is assumed that the `StdAir` log service has already been initialised with the proper log output stream by some other methods in the calling chain (for instance, when the [TRADEMGEN\\_Service](#) is itself being initialised by another library service such as `TVLSIM_Service`).

#### Parameters:

*stdair::STDAIR\_ServicePtr\_T* Handler on the `STDAIR_Service`.

*SEVMGR::SEVMGR\_ServicePtr\_T* Handler on the `SEVMGR_Service`.

*const* stdair::RandomSeed\_T& Seed for the random generation.

Definition at line 104 of file `TRADEMGEN_Service.cpp`.

**10.70.2.4 TRADEMGEN::TRADEMGEN\_Service::~~TRADEMGEN\_Service ()**

Destructor.

Definition at line 125 of file TRADEMGEN\_Service.cpp.

**10.70.3 Member Function Documentation****10.70.3.1 void TRADEMGEN::TRADEMGEN\_Service::parseAndLoad (const DemandFilePath &)**

Parse the demand input file.

The CSV file, describing the parameters of the demand to be generated for the simulator, is parsed and instantiated in memory accordingly.

**Parameters:**

*const* DemandFilePath& Filename of the input demand file.

Definition at line 241 of file TRADEMGEN\_Service.cpp.

References buildComplementaryLinks(), clonePersistentBom(), generateDemand(), TRADEMGEN::TRADEMGEN\_ServiceContext::getOwnStdairServiceFlag(), TRADEMGEN::TRADEMGEN\_ServiceContext::getPOSProbabilityMass(), TRADEMGEN::TRADEMGEN\_ServiceContext::getSEVMGR\_ServicePtr(), TRADEMGEN::TRADEMGEN\_ServiceContext::getSTDAIR\_Service(), and TRADEMGEN::TRADEMGEN\_ServiceContext::getUniformGenerator().

Referenced by TRADEMGEN::Trademgener::init(), and main().

**10.70.3.2 void TRADEMGEN::TRADEMGEN\_Service::buildSampleBom ()**

Build a sample BOM tree, made of a single DemandStream object.

As of now (March 2011), it corresponds to:

- Origin: SIN
- Destination: BKK
- Preferred departure date: 2011-02-14
- Preferred cabin: Y (Economy)
- POS distribution:
  - BKK: 30%
  - SIN: 70%
- Channel distribution:
  - Direct Offline: 10%
  - Direct Online: 30%
  - Indirect Offline: 40%
  - Indirect Online: 20%
- Trip type distribution:

- Outbound: 60%
  - Inbound: 20%
  - One-way: 20%
- Arrival pattern distribution:
  - 330 DTD: 0%
  - 40 DTD: 20%
  - 20 DTD: 60%
  - 1 DTD: 100%
- 15:0, 60:1
- Stay duration distribution:
  - 0 day: 10%
  - 1 day: 10%
  - 2 days: 15%
  - 3 days: 15%
  - 4 days: 15%
  - 5 days: 35%
- Frequent flyer distribution:
  - Platinum: 1%
  - Gold: 5%
  - Silver: 15%
  - Member: 30%
  - No card: 49%
- Preferred departure time (cumulative distribution):
  - 6am: 0%
  - 7am: 10%
  - 9am: 30%
  - 5pm: 40%
  - 7pm: 80%
  - 8pm: 95%
  - 10pm: 100%
- Value of time distribution:
  - 15 min: 0%
  - 60 min: 100%
- WTP: 200
- Number of requests: Normal ( $\mu = 10.0$ ,  $\text{std\_dev} = 1.0$ )
- Change fee: 20; Non refundable; Saturday night stay

Definition at line 310 of file TRADEMGEN\_Service.cpp.

References buildComplementaryLinks(), clonePersistentBom(), TRADEMGEN::TRADEMGEN\_ServiceContext::getOwnStdairServiceFlag(), TRADEMGEN::TRADEMGEN\_ServiceContext::getPOSProbabilityMass(), TRADEMGEN::TRADEMGEN\_ServiceContext::getSEVMGR\_ServicePtr(), TRADEMGEN::TRADEMGEN\_ServiceContext::getSTDAIR\_Service(), and TRADEMGEN::TRADEMGEN\_ServiceContext::getUniformGenerator().

Referenced by TRADEMGEN::Trademgener::init(), and main().

### 10.70.3.3 void TRADEMGEN::TRADEMGEN\_Service::clonePersistentBom ()

Clone the persistent BOM object.

Definition at line 382 of file TRADEMGEN\_Service.cpp.

References buildComplementaryLinks(), TRADEMGEN::TRADEMGEN\_ServiceContext::getOwnStdairServiceFlag(), and TRADEMGEN::TRADEMGEN\_ServiceContext::getSTDAIR\_Service().

Referenced by buildSampleBom(), and parseAndLoad().

### 10.70.3.4 void TRADEMGEN::TRADEMGEN\_Service::buildComplementaryLinks (stdair::BomRoot &)

Build all the complementary links in the given bom root object.

#### Note:

Do nothing for now.

Definition at line 419 of file TRADEMGEN\_Service.cpp.

Referenced by buildSampleBom(), clonePersistentBom(), and parseAndLoad().

### 10.70.3.5 stdair::BookingRequestStruct TRADEMGEN::TRADEMGEN\_Service::buildSampleBookingRequest (const bool isForCRS = false)

Build a sample booking request structure.

As of now (March 2011), the sample booking request is made of the following parameters:

- Return trip (inbound): LHR-SYD (POS: LHR, Channel: DN),
- Departing 10-JUN-2011 around 8:00, staying 7 days
- Requested on 15-MAY-2011 at 10:00
- Economy cabin, 3 persons, FF member
- WTP: 1000.0 EUR
- Dis-utility: 100.0 EUR/hour

As of now (March 2011), the CRS-related booking request is made of the following parameters:

- Return trip (inbound): SIN-BKK (POS: SIN, Channel: IN),
- Departing 30-JAN-2010 around 10:00, staying 7 days
- Requested on 22-JAN-2010 at 10:00

- Economy cabin, 3 persons, FF member
- WTP: 1000.0 EUR
- Dis-utility: 100.0 EUR/hour

**See also:**

stdair::CmdBomManager for more details.

**Parameters:**

**const** bool isForCRS Whether the sample booking request is for CRS.

**Returns:**

BookingRequestStruct& Sample booking request structure.

Definition at line 425 of file TRADEMGEN\_Service.cpp.

References TRADEMGEN::TRADEMGEN\_ServiceContext::getSTDAIR\_Service().

**10.70.3.6 void TRADEMGEN::TRADEMGEN\_Service::displayAirlineListFromDB () const**

Display the list of airlines, as held within the sample database.

Definition at line 535 of file TRADEMGEN\_Service.cpp.

Referenced by main().

**10.70.3.7 const stdair::Count\_T & TRADEMGEN::TRADEMGEN\_Service::getExpectedTotalNumberOfRequestsToBeGenerated () const**

Get the expected number of events/requests to be generated for all the demand streams.

Calls the SEvMgr service with the same name "getExpectedTotalNumberOfRequestsToBeGenerated", which computes that number.

**Note:**

That number usually corresponds to an expectation (i.e., the mean value of a random distribution). The actual number will be drawn when calling the [generateFirstRequests\(\)](#) method.

**Returns:**

const stdair::Count\_T& Expected number of events to be generated.

Definition at line 596 of file TRADEMGEN\_Service.cpp.

References TRADEMGEN::TRADEMGEN\_ServiceContext::getSEVMGR\_Service().

Referenced by generateDemand(), and TRADEMGEN::Trademgener::trademgen().

**10.70.3.8 const stdair::Count\_T & TRADEMGEN::TRADEMGEN\_Service::getActualTotalNumberOfRequestsToBeGenerated () const**

Get the actual number of events/requests to be generated for all the demand streams.

Calls the SEvMgr service with the same name "getActualTotalNumberOfRequestsToBeGenerated", which computes that number.



**Note:**

That number has been drawn when calling the [generateFirstRequests\(\)](#) method.

**Returns:**

const stdair::Count\_T& Expected number of events to be generated.

Definition at line 617 of file TRADEMGEN\_Service.cpp.

References TRADEMGEN::TRADEMGEN\_ServiceContext::getSEVMGR\_Service().

### 10.70.3.9 const bool TRADEMGEN::TRADEMGEN\_Service::stillHavingRequestsToBeGenerated (const stdair::DemandStreamKeyStr\_T &, stdair::ProgressStatusSet &, const stdair::DemandGenerationMethod &) const

Check whether enough requests have already been generated for the demand stream which corresponds to the given key.

**Parameters:**

**const DemandStreamKey&** A string identifying uniquely the demand stream (e.g., "SIN-HND 2010-Feb-08 Y").

**const stdair::DemandGenerationMethod&** States whether the demand generation must be performed following the method based on statistic orders. The alternative method, while more "intuitive", is also a sequential algorithm.

**Returns:**

bool Whether or not there are still events to be generated for that demand stream.

Definition at line 638 of file TRADEMGEN\_Service.cpp.

References TRADEMGEN::TRADEMGEN\_ServiceContext::getSEVMGR\_ServicePtr().

### 10.70.3.10 stdair::Count\_T TRADEMGEN::TRADEMGEN\_Service::generateFirstRequests (const stdair::DemandGenerationMethod &) const

Browse the list of demand streams and generate the first request of each stream.

**Parameters:**

**const stdair::DemandGenerationMethod&** States whether the demand generation must be performed following the method based on statistic orders. The alternative method, while more "intuitive", is also a sequential algorithm.

**Returns:**

stdair::Count\_T The expected total number of events to be generated

Definition at line 663 of file TRADEMGEN\_Service.cpp.

References TRADEMGEN::TRADEMGEN\_ServiceContext::getSEVMGR\_ServicePtr(), and TRADEMGEN::TRADEMGEN\_ServiceContext::getUniformGenerator().

Referenced by generateDemand(), and TRADEMGEN::Trademgener::trademgen().

### 10.70.3.11 stdair::BookingRequestPtr\_T TRADEMGEN::TRADEMGEN\_Service::generateNextRequest (const stdair::DemandStreamKeyStr\_T &, const stdair::DemandGenerationMethod &) const

Generate a request with the demand stream which corresponds to the given key.

#### Parameters:

**const DemandStreamKey** & A string identifying uniquely the demand stream (e.g., "SIN-HND 2010-Feb-08 Y").

**const stdair::DemandGenerationMethod** & States whether the demand generation must be performed following the method based on statistic orders. The alternative method, while more "intuitive", is also a sequential algorithm.

#### Returns:

stdair::BookingRequestPtr\_T (Boost) shared pointer on the booking request structure, which has just been created.

Definition at line 689 of file TRADEMGEN\_Service.cpp.

References TRADEMGEN::TRADEMGEN\_ServiceContext::getSEVMGR\_ServicePtr(), and TRADEMGEN::TRADEMGEN\_ServiceContext::getUniformGenerator().

Referenced by generateDemand(), and TRADEMGEN::Trademgener::trademgen().

### 10.70.3.12 bool TRADEMGEN::TRADEMGEN\_Service::hasDemandStream (const stdair::DemandStreamKeyStr\_T &) const

States whether a demand stream with the given key is used to generate demand.

#### Parameters:

**const DemandStreamKey** & A string identifying uniquely the demand stream (e.g., "SIN-HND 2010-Feb-08 Y").

Definition at line 868 of file TRADEMGEN\_Service.cpp.

References TRADEMGEN::TRADEMGEN\_ServiceContext::getSEVMGR\_ServicePtr().

### 10.70.3.13 stdair::ProgressStatusSet TRADEMGEN::TRADEMGEN\_Service::popEvent (stdair::EventStruct &) const

Pop the next coming (in time) event, and remove it from the event queue thanks to the SEvMgr service.

- The next coming (in time) event corresponds to the event having the earliest date-time stamp. In other words, it is the first/front element of the event queue.
- That (first) event/element is then removed from the event queue
- The progress status is updated for the corresponding demand stream.

#### Returns:

stdair::EventStruct A copy of the event structure, which comes first in time from within the event queue.

Definition at line 713 of file TRADEMGEN\_Service.cpp.

References TRADEMGEN::TRADEMGEN\_ServiceContext::getSEVMGR\_Service().

Referenced by generateDemand(), and TRADEMGEN::Trademgener::trademgen().

#### 10.70.3.14 bool TRADEMGEN::TRADEMGEN\_Service::isQueueDone () const

States whether the event queue has reached the end.

Calls the SEvMgr service with the same name "isQueueDone", which states whether the event queue has reached the end.

For now, that method states whether the event queue is empty.

Definition at line 729 of file TRADEMGEN\_Service.cpp.

References TRADEMGEN::TRADEMGEN\_ServiceContext::getSEVMGR\_Service().

Referenced by generateDemand(), and TRADEMGEN::Trademgener::trademgen().

#### 10.70.3.15 bool TRADEMGEN::TRADEMGEN\_Service::generateCancellation (const stdair::TravelSolutionStruct &, const stdair::PartySize\_T &, const stdair::DateTime\_T &, const stdair::Date\_T &) const

Generate the potential cancellation event.

Definition at line 749 of file TRADEMGEN\_Service.cpp.

References TRADEMGEN::TRADEMGEN\_ServiceContext::getSEVMGR\_Service(), and TRADEMGEN::TRADEMGEN\_ServiceContext::getUniformGenerator().

#### 10.70.3.16 void TRADEMGEN::TRADEMGEN\_Service::reset () const

Reset the context of the demand streams for another demand generation without having to reparse the demand input file.

Definition at line 811 of file TRADEMGEN\_Service.cpp.

References TRADEMGEN::TRADEMGEN\_ServiceContext::getSEVMGR\_ServicePtr(), and TRADEMGEN::TRADEMGEN\_ServiceContext::getUniformGenerator().

Referenced by generateDemand(), and TRADEMGEN::Trademgener::trademgen().

#### 10.70.3.17 const stdair::ProgressStatus & TRADEMGEN::TRADEMGEN\_Service::getProgress-Status () const

Get the overall progress status (for the whole event queue).

Definition at line 832 of file TRADEMGEN\_Service.cpp.

References TRADEMGEN::TRADEMGEN\_ServiceContext::getSEVMGR\_ServicePtr().

#### 10.70.3.18 const stdair::ProgressStatus & TRADEMGEN::TRADEMGEN\_Service::getProgress-Status (const stdair::EventType::EN\_EventType &) const

Get the progress status for the given event type (e.g., booking request, optimisation notification, schedule change, break point).

Definition at line 850 of file TRADEMGEN\_Service.cpp.

References TRADEMGEN::TRADEMGEN\_ServiceContext::getSEVMGR\_ServicePtr().

### 10.70.3.19 std::string TRADEMGEN::TRADEMGEN\_Service::jsonHandler (const stdair::JSONString &) const

Dispatch the JSon command string to the SEvMgr service. (Only SEvMgr has json export commands for now).

#### Parameters:

*const* stdair::JSONString& Input string which contained the JSon command string.

#### Returns:

std::string Output string in which the asking objects are logged/dumped with a JSon format.

Definition at line 447 of file TRADEMGEN\_Service.cpp.

References TRADEMGEN::TRADEMGEN\_ServiceContext::getSEVMGR\_ServicePtr().

### 10.70.3.20 std::string TRADEMGEN::TRADEMGEN\_Service::csvDisplay () const

Recursively display (dump in the returned string) the objects of the BOM tree.

#### Returns:

std::string Output string in which the BOM tree is logged/dumped.

Definition at line 468 of file TRADEMGEN\_Service.cpp.

References TRADEMGEN::BomDisplay::csvDisplay(), and TRADEMGEN::TRADEMGEN\_ServiceContext::getSEVMGR\_ServicePtr().

Referenced by generateDemand(), and TRADEMGEN::Trademgener::trademgen().

### 10.70.3.21 std::string TRADEMGEN::TRADEMGEN\_Service::list () const

Display (dump in the returned string) the event list of the event queue.

#### Returns:

std::string Output string in which the events are logged/dumped.

Definition at line 490 of file TRADEMGEN\_Service.cpp.

References TRADEMGEN::TRADEMGEN\_ServiceContext::getSEVMGR\_ServicePtr().

### 10.70.3.22 std::string TRADEMGEN::TRADEMGEN\_Service::list (const stdair::EventType::EN\_EventType &) const

Display (dump in the returned string) the events with the given type

#### Returns:

std::string Output string in which the events are logged/dumped.

Definition at line 513 of file TRADEMGEN\_Service.cpp.

References TRADEMGEN::TRADEMGEN\_ServiceContext::getSEVMGR\_ServicePtr().

**10.70.3.23 std::string TRADEMGEN::TRADEMGEN\_Service::displayDemandStream () const**

Display (dump in the returned string) the demand streams

**Returns:**

std::string Output string in which the demand streams are logged/dumped.

Definition at line 885 of file TRADEMGEN\_Service.cpp.

References TRADEMGEN::TRADEMGEN\_ServiceContext::getSEVMGR\_ServicePtr().

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

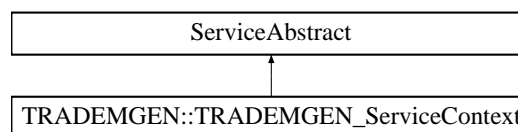
- trademgen/[TRADEMGEN\\_Service.hpp](#)
- trademgen/service/[TRADEMGEN\\_Service.cpp](#)

**10.71 TRADEMGEN::TRADEMGEN\_ServiceContext Class Reference**

Class holding the context of the Trademgen services.

```
#include <trademgen/service/TRADEMGEN_ServiceContext.hpp>
```

Inheritance diagram for TRADEMGEN::TRADEMGEN\_ServiceContext::

**Friends**

- class [TRADEMGEN\\_Service](#)
- class [FacTRADEMGENServiceContext](#)

**10.71.1 Detailed Description**

Class holding the context of the Trademgen services.

Definition at line 34 of file TRADEMGEN\_ServiceContext.hpp.

**10.71.2 Friends And Related Function Documentation****10.71.2.1 friend class [TRADEMGEN\\_Service](#) [friend]**

The [TRADEMGEN\\_Service](#) class should be the sole class to get access to ServiceContext content: general users do not want to bother with a context interface.

Definition at line 40 of file TRADEMGEN\_ServiceContext.hpp.

### 10.71.2.2 friend class [FacTRADEMGENServiceContext](#) [friend]

Definition at line 41 of file TRADEMGEN\_ServiceContext.hpp.

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

- trademgen/service/[TRADEMGEN\\_ServiceContext.hpp](#)
- trademgen/service/[TRADEMGEN\\_ServiceContext.cpp](#)

## 10.72 TRADEMGEN::Trademgener Struct Reference

Wrapper structure around the C++ API, so as to expose a Python API.

### Public Member Functions

- std::string [trademgen](#) (const [NbOfRuns\\_T](#) &iNbOfRuns, const std::string &iDemandGeneration-MethodString)
- [Trademgener](#) ()
- [Trademgener](#) (const [Trademgener](#) &iTrademgener)
- [~Trademgener](#) ()
- bool [init](#) (const std::string &iLogFilePath, const stdair::RandomSeed\_T &iRandomSeed, const bool isBuiltin, const stdair::Filename\_T &iDemandInputFilename)

### 10.72.1 Detailed Description

Wrapper structure around the C++ API, so as to expose a Python API.

Definition at line 70 of file pytrademgen.cpp.

### 10.72.2 Constructor & Destructor Documentation

#### 10.72.2.1 TRADEMGEN::Trademgener::Trademgener () [inline]

Default constructor.

Definition at line 266 of file pytrademgen.cpp.

#### 10.72.2.2 TRADEMGEN::Trademgener::Trademgener (const [Trademgener](#) & iTrademgener) [inline]

Default copy constructor.

Definition at line 270 of file pytrademgen.cpp.

#### 10.72.2.3 TRADEMGEN::Trademgener::~~Trademgener () [inline]

Default constructor.

Definition at line 276 of file pytrademgen.cpp.

### 10.72.3 Member Function Documentation

#### 10.72.3.1 `std::string TRADEMGEN::Trademgener::trademgen (const NbOfRuns\_T & iNbOfRuns, const std::string & iDemandGenerationMethodString)` `[inline]`

Wrapper around the travel demand generation use case.

Definition at line 76 of file `pytrademgen.cpp`.

References `TRADEMGEN::TRADEMGEN_Service::csvDisplay()`, `TRADEMGEN::TRADEMGEN_Service::generateFirstRequests()`, `TRADEMGEN::TRADEMGEN_Service::generateNextRequest()`, `TRADEMGEN::TRADEMGEN_Service::getExpectedTotalNumberOfRequestsToBeGenerated()`, `TRADEMGEN::TRADEMGEN_Service::isQueueDone()`, `TRADEMGEN::TRADEMGEN_Service::popEvent()`, `TRADEMGEN::TRADEMGEN_Service::reset()`, and `TRADEMGEN::stat_display()`.

Referenced by `BOOST_PYTHON_MODULE()`.

#### 10.72.3.2 `bool TRADEMGEN::Trademgener::init (const std::string & iLogFilepath, const stdair::RandomSeed_T & iRandomSeed, const bool isBuiltin, const stdair::Filename_T & iDemandInputFilename)` `[inline]`

Wrapper around the search use case.

Definition at line 284 of file `pytrademgen.cpp`.

References `TRADEMGEN::TRADEMGEN_Service::buildSampleBom()`, `TRADEMGEN::TRADEMGEN_Service::parseAndLoad()`, and

Referenced by `BOOST_PYTHON_MODULE()`.

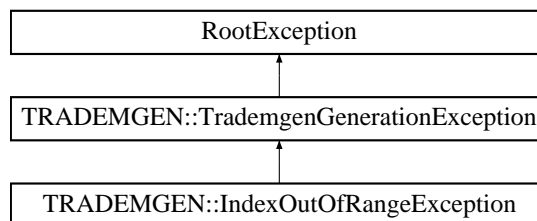
The documentation for this struct was generated from the following file:

- `trademgen/python/pytrademgen.cpp`

## 10.73 TRADEMGEN::TrademgenGenerationException Class Reference

```
#include <trademgen/TRADEMGEN_Exceptions.hpp>
```

Inheritance diagram for `TRADEMGEN::TrademgenGenerationException`:



### Public Member Functions

- [TrademgenGenerationException](#) (const std::string &*iWhat*)

### 10.73.1 Detailed Description

Root exception for the TraDemGen component

Definition at line 18 of file TRADEMGEN\_Exceptions.hpp.

### 10.73.2 Constructor & Destructor Documentation

#### 10.73.2.1 TRADEMGEN::TrademgenGenerationException::TrademgenGenerationException (const std::string & *iWhat*) [inline]

Constructor.

Definition at line 23 of file TRADEMGEN\_Exceptions.hpp.

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

- [trademgen/TRADEMGEN\\_Exceptions.hpp](#)



## 11 TraDemGen File Documentation

- 11.1 doc/local/authors.doc File Reference
- 11.2 doc/local/codingrules.doc File Reference
- 11.3 doc/local/copyright.doc File Reference
- 11.4 doc/local/documentation.doc File Reference
- 11.5 doc/local/features.doc File Reference
- 11.6 doc/local/help\_wanted.doc File Reference
- 11.7 doc/local/howto\_release.doc File Reference
- 11.8 doc/local/index.doc File Reference
- 11.9 doc/local/installation.doc File Reference
- 11.10 doc/local/linking.doc File Reference
- 11.11 doc/local/test.doc File Reference
- 11.12 doc/local/users\_guide.doc File Reference
- 11.13 doc/local/verification.doc File Reference
- 11.14 doc/tutorial/tutorial.doc File Reference
- 11.15 test/trademgen/DemandGenerationTestSuite.cpp File Reference
- 11.16 test/trademgen/DemandGenerationTestSuite.hpp File Reference

```
#include <iosfwd>
```

```
#include <cppunit/extensions/HelperMacros.h>
```

### Classes

- class [DemandGenerationTestSuite](#)

### Functions

- [CPPUNIT\\_TEST\\_SUITE\\_REGISTRATION](#) ([DemandGenerationTestSuite](#))

#### 11.16.1 Function Documentation

##### 11.16.1.1 CPPUNIT\_TEST\_SUITE\_REGISTRATION ([DemandGenerationTestSuite](#))

## 11.17 test/trademgen/generateEvents.cpp File Reference

```
#include <cassert>
#include <string>
#include <map>
#include <iostream>
#include <sstream>
#include <test/trademgen/EventStream.hpp>
#include <test/trademgen/CategoricalAttribute.hpp>
```

### Functions

- int [main](#) (int argc, char \*const argv[ ])

#### 11.17.1 Function Documentation

##### 11.17.1.1 int main (int *argc*, char \*const *argv*[ ])

Definition at line 12 of file generateEvents.cpp.

## 11.18 trademgen/basic/BasConst.cpp File Reference

```
#include <stdair/basic/BasConst_General.hpp>
#include <trademgen/basic/BasConst_TRADEMGEN_Service.hpp>
#include <trademgen/basic/BasConst_DemandGeneration.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

### Functions

- stdair::BaseGenerator\_T [TRADEMGEN::DEFAULT\\_BASE\\_GENERATOR](#) (stdair::DEFAULT\_RANDOM\_SEED)
- stdair::UniformGenerator\_T [TRADEMGEN::DEFAULT\\_UNIFORM\\_GENERATOR](#) (DEFAULT\_BASE\_GENERATOR, [DEFAULT\\_UNIFORM\\_REAL\\_DISTRIBUTION](#))

### Variables

- const [POSProbabilityMassFunction\\_T](#) [TRADEMGEN::DEFAULT\\_POS\\_PROBALILITY\\_MASS](#)
- const stdair::FloatDuration\_T [TRADEMGEN::DEFAULT\\_LAST\\_LOWER\\_BOUND\\_ARRIVAL\\_PATTERN](#) = -1
- const [FRAT5Pattern\\_T](#) [TRADEMGEN::DEFAULT\\_FRAT5\\_PATTERN](#) = DefaultMap::createFRAT5Pattern()
- const double [TRADEMGEN::DEFAULT\\_MAX\\_ADVANCE\\_PURCHASE](#) = 330.0
- const stdair::UniformDistribution\_T [TRADEMGEN::DEFAULT\\_UNIFORM\\_REAL\\_DISTRIBUTION](#)

## 11.19 trademgen/basic/BasConst\_DemandGeneration.hpp File Reference

```
#include <string>
#include <stdair/stdair_maths_types.hpp>
#include <stdair/stdair_date_time_types.hpp>
#include <trademgen/basic/DemandCharacteristicsTypes.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

### Classes

- struct [TRADEMGEN::DefaultMap](#)

### Variables

- const [POSProbabilityMassFunction\\_T TRADEMGEN::DEFAULT\\_POS\\_PROBALILITY\\_MASS](#)
- const [FRAT5Pattern\\_T TRADEMGEN::DEFAULT\\_FRAT5\\_PATTERN](#)
- const [stdair::FloatDuration\\_T TRADEMGEN::DEFAULT\\_LAST\\_LOWER\\_BOUND\\_ARRIVAL\\_PATTERN](#)
- const double [TRADEMGEN::DEFAULT\\_MAX\\_ADVANCE\\_PURCHASE](#)
- [stdair::BaseGenerator\\_T TRADEMGEN::DEFAULT\\_BASE\\_GENERATOR](#)
- [stdair::UniformGenerator\\_T TRADEMGEN::DEFAULT\\_UNIFORM\\_GENERATOR](#)
- const [stdair::UniformDistribution\\_T TRADEMGEN::DEFAULT\\_UNIFORM\\_REAL\\_DISTRIBUTION](#)

## 11.20 trademgen/basic/BasConst\_TRADEMGEN\_Service.hpp File Reference

```
#include <string>
```

### Namespaces

- namespace [TRADEMGEN](#)

## 11.21 trademgen/basic/BasParserTypes.hpp File Reference

```
#include <string>
#include <boost/spirit/home/classic/core.hpp>
#include <boost/spirit/home/classic/utility/loops.hpp>
#include <boost/spirit/home/classic/utility/chset.hpp>
#include <boost/spirit/home/classic/utility/confix.hpp>
#include <boost/spirit/home/classic/iterator/file_iterator.hpp>
```

## Namespaces

- namespace [TRADEMGEN](#)

## Typedefs

- typedef char [TRADEMGEN::char\\_t](#)
- typedef boost::spirit::classic::file\_iterator< [char\\_t](#) > [TRADEMGEN::iterator\\_t](#)
- typedef boost::spirit::classic::scanner< [iterator\\_t](#) > [TRADEMGEN::scanner\\_t](#)
- typedef boost::spirit::classic::rule< [scanner\\_t](#) > [TRADEMGEN::rule\\_t](#)
- typedef boost::spirit::classic::int\_parser< unsigned int, 10, 1, 1 > [TRADEMGEN::int1\\_p\\_t](#)
- typedef boost::spirit::classic::uint\_parser< unsigned int, 10, 2, 2 > [TRADEMGEN::uint2\\_p\\_t](#)
- typedef boost::spirit::classic::uint\_parser< unsigned int, 10, 1, 2 > [TRADEMGEN::uint1\\_2\\_p\\_t](#)
- typedef boost::spirit::classic::uint\_parser< unsigned int, 10, 1, 3 > [TRADEMGEN::uint1\\_3\\_p\\_t](#)
- typedef boost::spirit::classic::uint\_parser< unsigned int, 10, 4, 4 > [TRADEMGEN::uint4\\_p\\_t](#)
- typedef boost::spirit::classic::uint\_parser< unsigned int, 10, 1, 4 > [TRADEMGEN::uint1\\_4\\_p\\_t](#)
- typedef boost::spirit::classic::chset< [char\\_t](#) > [TRADEMGEN::chset\\_t](#)
- typedef boost::spirit::classic::impl::loop\_traits< [chset\\_t](#), unsigned int, unsigned int >::type [TRADEMGEN::repeat\\_p\\_t](#)
- typedef boost::spirit::classic::bounded< [uint2\\_p\\_t](#), unsigned int > [TRADEMGEN::bounded2\\_p\\_t](#)
- typedef boost::spirit::classic::bounded< [uint1\\_2\\_p\\_t](#), unsigned int > [TRADEMGEN::bounded1\\_2\\_p\\_t](#)
- typedef boost::spirit::classic::bounded< [uint1\\_3\\_p\\_t](#), unsigned int > [TRADEMGEN::bounded1\\_3\\_p\\_t](#)
- typedef boost::spirit::classic::bounded< [uint4\\_p\\_t](#), unsigned int > [TRADEMGEN::bounded4\\_p\\_t](#)
- typedef boost::spirit::classic::bounded< [uint1\\_4\\_p\\_t](#), unsigned int > [TRADEMGEN::bounded1\\_4\\_p\\_t](#)

## 11.22 trademgen/basic/CategoricalAttribute.hpp File Reference

```
#include <map>
#include <iosfwd>
#include <stdair/STDAIR_Types.hpp>
#include <stdair/basic/DictionaryManager.hpp>
```

## Namespaces

- namespace [stdair](#)

## Classes

- struct [stdair::CategoricalAttribute< T >](#)

*Class modeling the distribution of values that can be taken by a categorical attribute.*

## 11.23 trademgen/basic/CategoricalAttributeLite.hpp File Reference

```
#include <cassert>
#include <sstream>
#include <string>
#include <vector>
#include <map>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/service/Logger.hpp>
#include <trademgen/TRADEMGEN_Exceptions.hpp>
#include <trademgen/basic/DictionaryManager.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

### Classes

- struct [TRADEMGEN::CategoricalAttribute< T >](#)  
*Class modeling the distribution of values that can be taken by a categorical attribute.*

## 11.24 trademgen/basic/ContinuousAttribute.hpp File Reference

```
#include <string>
#include <map>
#include <stdair/stdair_date_time_types.hpp>
#include <stdair/service/Logger.hpp>
#include <trademgen/basic/DictionaryManager.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

### Classes

- struct [TRADEMGEN::ContinuousAttribute< T >](#)

## 11.25 trademgen/basic/ContinuousAttributeLite.hpp File Reference

```
#include <cassert>
#include <iosfwd>
#include <string>
```

```
#include <vector>
#include <map>
#include <stdair/stdair_basic_types.hpp>
#include <trademgen/TRADEMGEN_Exceptions.hpp>
#include <trademgen/basic/DictionaryManager.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

### Classes

- struct [TRADEMGEN::ContinuousAttributeLite< T >](#)  
*Class modeling the distribution of values that can be taken by a continuous attribute.*

## 11.26 trademgen/basic/DemandCharacteristics.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/stdair_basic_types.hpp>
#include <trademgen/basic/BasConst_DemandGeneration.hpp>
#include <trademgen/basic/DemandCharacteristics.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

## 11.27 trademgen/basic/DemandCharacteristics.hpp File Reference

```
#include <string>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_date_time_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
#include <trademgen/basic/DemandCharacteristicsTypes.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

### Classes

- struct [TRADEMGEN::DemandCharacteristics](#)

*Class modeling the characteristics of a demand type.*

## 11.28 trademgen/basic/DemandCharacteristicsTypes.hpp File Reference

```
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_date_time_types.hpp>
#include <stdair/stdair_demand_types.hpp>
#include <trademgen/basic/ContinuousAttributeLite.hpp>
#include <trademgen/basic/CategoricalAttributeLite.hpp>
```

### Namespaces

- namespace [TRADEMG](#)

### Typedefs

- typedef ContinuousAttributeLite< stdair::FloatDuration\_T > [TRADEMG::ContinuousFloatDuration\\_T](#)
- typedef ContinuousFloatDuration\_T::ContinuousDistribution\_T [TRADEMG::ArrivalPatternCumulativeDistribution\\_T](#)
- typedef CategoricalAttributeLite< stdair::AirportCode\_T > [TRADEMG::POSProbabilityMass\\_T](#)
- typedef POSProbabilityMass\_T::ProbabilityMassFunction\_T [TRADEMG::POSProbabilityMassFunction\\_T](#)
- typedef CategoricalAttributeLite< stdair::ChannelLabel\_T > [TRADEMG::ChannelProbabilityMass\\_T](#)
- typedef ChannelProbabilityMass\_T::ProbabilityMassFunction\_T [TRADEMG::ChannelProbabilityMassFunction\\_T](#)
- typedef CategoricalAttributeLite< stdair::TripType\_T > [TRADEMG::TripTypeProbabilityMass\\_T](#)
- typedef TripTypeProbabilityMass\_T::ProbabilityMassFunction\_T [TRADEMG::TripTypeProbabilityMassFunction\\_T](#)
- typedef CategoricalAttributeLite< stdair::DayDuration\_T > [TRADEMG::StayDurationProbabilityMass\\_T](#)
- typedef StayDurationProbabilityMass\_T::ProbabilityMassFunction\_T [TRADEMG::StayDurationProbabilityMassFunction\\_T](#)
- typedef CategoricalAttributeLite< stdair::FrequentFlyer\_T > [TRADEMG::FrequentFlyerProbabilityMass\\_T](#)
- typedef FrequentFlyerProbabilityMass\_T::ProbabilityMassFunction\_T [TRADEMG::FrequentFlyerProbabilityMassFunction\\_T](#)
- typedef ContinuousAttributeLite< stdair::IntDuration\_T > [TRADEMG::PreferredDepartureTimeCumulativeDistribution\\_T](#)
- typedef PreferredDepartureTimeCumulativeDistribution\_T::ContinuousDistribution\_T [TRADEMG::PreferredDepartureTimeContinuousDistribution\\_T](#)
- typedef ContinuousAttributeLite< stdair::PriceValue\_T > [TRADEMG::ValueOfTimeCumulativeDistribution\\_T](#)
- typedef ValueOfTimeCumulativeDistribution\_T::ContinuousDistribution\_T [TRADEMG::ValueOfTimeContinuousDistribution\\_T](#)

- typedef ContinuousAttributeLite< stdair::RealNumber\_T > [TRADEMGEN::CumulativeDistribution\\_T](#)
- typedef CumulativeDistribution\_T::ContinuousDistribution\_T [TRADEMGEN::FRAT5Pattern\\_T](#)

## 11.29 trademgen/basic/DemandDistribution.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/stdair_date_time_types.hpp>
#include <trademgen/basic/DemandDistribution.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

## 11.30 trademgen/basic/DemandDistribution.hpp File Reference

```
#include <string>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
#include <trademgen/basic/ContinuousAttribute.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

### Classes

- struct [TRADEMGEN::DemandDistribution](#)  
*Class modeling the distribution of a demand type.*

## 11.31 trademgen/basic/DictionaryManager.cpp File Reference

```
#include <trademgen/basic/DictionaryManager.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

## 11.32 trademgen/basic/DictionaryManager.hpp File Reference

```
#include <stdair/stdair_maths_types.hpp>
```



## Namespaces

- namespace [TRADEMGEN](#)

## Classes

- class [TRADEMGEN::DictionaryManager](#)  
*Class wrapper of dictionary business methods.*

## Typedefs

- typedef stdair::Probability\_T [TRADEMGEN::DictionaryKey\\_T](#)

## 11.33 trademgen/basic/RandomGenerationContext.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <trademgen/basic/RandomGenerationContext.hpp>
```

## Namespaces

- namespace [TRADEMGEN](#)

## 11.34 trademgen/basic/RandomGenerationContext.hpp File Reference

```
#include <iosfwd>
#include <string>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_maths_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
```

## Namespaces

- namespace [TRADEMGEN](#)

## Classes

- struct [TRADEMGEN::RandomGenerationContext](#)

## 11.35 trademgen/batches/trademgen\_generateDemand.cpp File Reference

```
#include <cassert>
#include <sstream>
```

```

#include <fstream>
#include <vector>
#include <list>
#include <string>
#include <boost/tokenizer.hpp>
#include <boost/program_options.hpp>
#include <boost/accumulators/accumulators.hpp>
#include <boost/accumulators/statistics.hpp>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/basic/BasConst_General.hpp>
#include <stdair/basic/ProgressStatusSet.hpp>
#include <stdair/basic/DemandGenerationMethod.hpp>
#include <stdair/bom/EventStruct.hpp>
#include <stdair/bom/BookingRequestStruct.hpp>
#include <stdair/bom/BomDisplay.hpp>
#include <stdair/service/Logger.hpp>
#include <trademgen/TRADEMGEN_Service.hpp>
#include <trademgen/config/trademgen-paths.hpp>

```

## Typedefs

- typedef unsigned int [NbOfRuns\\_T](#)
- typedef ba::accumulator\_set< double, ba::stats< ba::tag::min, ba::tag::max, ba::tag::mean(ba::immediate), ba::tag::sum, ba::tag::variance > > [stat\\_acc\\_type](#)

## Functions

- const stdair::Filename\_T [K\\_TRADEMGEN\\_DEFAULT\\_LOG\\_FILENAME](#) ("trademgen\_generateDemand.log")
- const stdair::Filename\_T [K\\_TRADEMGEN\\_DEFAULT\\_INPUT\\_FILENAME](#) (STDAIR\_SAMPLE\_DIR"/demand01.csv")
- const stdair::Filename\_T [K\\_TRADEMGEN\\_DEFAULT\\_OUTPUT\\_FILENAME](#) ("request.csv")
- void [stat\\_display](#) (std::ostream &oStream, const [stat\\_acc\\_type](#) &iStatAcc)
- template<class T> std::ostream & [operator<<](#) (std::ostream &os, const std::vector< T > &v)
- int [readConfiguration](#) (int argc, char \*argv[], bool &ioIsBuiltin, stdair::RandomSeed\_T &ioRandomSeed, [NbOfRuns\\_T](#) &ioRandomRuns, stdair::Filename\_T &ioInputFilename, stdair::Filename\_T &ioOutputFilename, stdair::Filename\_T &ioLogFilename, stdair::DemandGenerationMethod &ioDemandGenerationMethod)
- void [generateDemand](#) (TRADEMGEN::TRADEMGEN\_Service &ioTrademgenService, const stdair::Filename\_T &ioOutputFilename, const [NbOfRuns\\_T](#) &iNbOfRuns, const stdair::DemandGenerationMethod &iDemandGenerationMethod)
- int [main](#) (int argc, char \*argv[])

## Variables

- const stdair::DemandGenerationMethod [K\\_TRADEMGEN\\_DEFAULT\\_DEMAND\\_GENERATION\\_METHOD](#)
- const char [K\\_TRADEMGEN\\_DEFAULT\\_DEMAND\\_GENERATION\\_METHOD\\_CHAR](#)
- const stdair::RandomSeed\_T [K\\_TRADEMGEN\\_DEFAULT\\_RANDOM\\_SEED](#)
- const [NbOfRuns\\_T](#) [K\\_TRADEMGEN\\_DEFAULT\\_RANDOM\\_DRAWS](#) = 1
- const bool [K\\_TRADEMGEN\\_DEFAULT\\_BUILT\\_IN\\_INPUT](#) = false
- const int [K\\_TRADEMGEN\\_EARLY\\_RETURN\\_STATUS](#) = 99

### 11.35.1 Typedef Documentation

#### 11.35.1.1 typedef unsigned int [NbOfRuns\\_T](#)

Definition at line 38 of file trademgen\_generateDemand.cpp.

#### 11.35.1.2 typedef [ba::accumulator\\_set<double, ba::stats<ba::tag::min, ba::tag::max, ba::tag::mean \(ba::immediate\), ba::tag::sum, ba::tag::variance>>](#) [stat\\_acc\\_type](#)

Type definition to gather statistics.

Definition at line 47 of file trademgen\_generateDemand.cpp.

### 11.35.2 Function Documentation

#### 11.35.2.1 const stdair::Filename\_T [K\\_TRADEMGEN\\_DEFAULT\\_LOG\\_FILENAME](#) ("trademgen\_generateDemand.log")

Default name and location for the log file.

Referenced by readConfiguration().

#### 11.35.2.2 const stdair::Filename\_T [K\\_TRADEMGEN\\_DEFAULT\\_INPUT\\_FILENAME](#) (STDAIR\_SAMPLE\_DIR"/demand01.csv")

Default name and location for the (CSV) input file.

Referenced by readConfiguration().

#### 11.35.2.3 const stdair::Filename\_T [K\\_TRADEMGEN\\_DEFAULT\\_OUTPUT\\_FILENAME](#) ("request.csv")

Default name and location for the (CSV) output file.

Referenced by readConfiguration().

#### 11.35.2.4 void stat\_display (std::ostream & *oStream*, const [stat\\_acc\\_type](#) & *iStatAcc*)

Display the statistics held by the dedicated accumulator.

Definition at line 105 of file trademgen\_generateDemand.cpp.

Referenced by generateDemand().

### 11.35.2.5 `template<class T> std::ostream& operator<< (std::ostream & os, const std::vector< T > & v)`

Definition at line 127 of file trademgen\_generateDemand.cpp.

### 11.35.2.6 `int readConfiguration (int argc, char * argv[], bool & ioIsBuiltin, stdair::RandomSeed_T & ioRandomSeed, NbOfRuns_T & ioRandomRuns, stdair::Filename_T & ioInputFilename, stdair::Filename_T & ioOutputFilename, stdair::Filename_T & ioLogFilename, stdair::DemandGenerationMethod & ioDemandGenerationMethod)`

Read and parse the command line options.

Definition at line 136 of file trademgen\_generateDemand.cpp.

References `K_TRADEMGEN_DEFAULT_DEMAND_GENERATION_METHOD_CHAR`, `K_TRADEMGEN_DEFAULT_INPUT_FILENAME()`, `K_TRADEMGEN_DEFAULT_LOG_FILENAME()`, and `K_TRADEMGEN_DEFAULT_OUTPUT_FILENAME()`.

Referenced by `main()`.

### 11.35.2.7 `void generateDemand (TRADEMGEN::TRADEMGEN_Service & ioTrademgenService, const stdair::Filename_T & ioOutputFilename, const NbOfRuns_T & iNbOfRuns, const stdair::DemandGenerationMethod & iDemandGenerationMethod)`

Definition at line 276 of file trademgen\_generateDemand.cpp.

References `TRADEMGEN::TRADEMGEN_Service::csvDisplay()`, `TRADEMGEN::TRADEMGEN_Service::generateFirstRequests()`, `TRADEMGEN::TRADEMGEN_Service::generateNextRequest()`, `TRADEMGEN::TRADEMGEN_Service::getExpectedTotalNumberOfRequestsToBeGenerated()`, `TRADEMGEN::TRADEMGEN_Service::isQueueDone()`, `TRADEMGEN::TRADEMGEN_Service::popEvent()`, `TRADEMGEN::TRADEMGEN_Service::reset()`, and `stat_display()`.

Referenced by `main()`, and `TRADEMGEN::TRADEMGEN_Service::parseAndLoad()`.

### 11.35.2.8 `int main (int argc, char * argv[])`

Definition at line 420 of file trademgen\_generateDemand.cpp.

References `TRADEMGEN::TRADEMGEN_Service::buildSampleBom()`, `generateDemand()`, `K_TRADEMGEN_DEFAULT_DEMAND_GENERATION_METHOD`, `TRADEMGEN::TRADEMGEN_Service::parseAndLoad()`, and `readConfiguration()`.

## 11.35.3 Variable Documentation

### 11.35.3.1 `const stdair::DemandGenerationMethod K_TRADEMGEN_DEFAULT_DEMAND_GENERATION_METHOD`

**Initial value:**

```
stdair::DemandGenerationMethod::POI_PRO
```

Default demand generation method: Poisson Process.

Definition at line 70 of file trademgen\_generateDemand.cpp.

Referenced by `main()`.

**11.35.3.2 const char [K\\_TRADEMGEN\\_DEFAULT\\_DEMAND\\_GENERATION\\_METHOD\\_CHAR](#)****Initial value:**

```
K_TRADEMGEN_DEFAULT_DEMAND_GENERATION_METHOD.getMethodAsChar()
```

Default demand generation method name: 'P' for Poisson Process.

Definition at line 76 of file trademgen\_generateDemand.cpp.

Referenced by readConfiguration().

**11.35.3.3 const stdair::RandomSeed\_T [K\\_TRADEMGEN\\_DEFAULT\\_RANDOM\\_SEED](#)****Initial value:**

```
stdair::DEFAULT_RANDOM_SEED
```

Default random generation seed (e.g., 120765987).

Definition at line 82 of file trademgen\_generateDemand.cpp.

Referenced by readConfiguration().

**11.35.3.4 const [NbOfRuns\\_T](#) [K\\_TRADEMGEN\\_DEFAULT\\_RANDOM\\_DRAWS](#) = 1**

Default number of random draws to be generated (best if over 100).

Definition at line 88 of file trademgen\_generateDemand.cpp.

**11.35.3.5 const bool [K\\_TRADEMGEN\\_DEFAULT\\_BUILT\\_IN\\_INPUT](#) = false**

Default for the input type. It can be either built-in or provided by an input file. That latter must then be given with the -i option.

Definition at line 94 of file trademgen\_generateDemand.cpp.

Referenced by readConfiguration().

**11.35.3.6 const int [K\\_TRADEMGEN\\_EARLY\\_RETURN\\_STATUS](#) = 99**

Early return status (so that it can be differentiated from an error).

Definition at line 99 of file trademgen\_generateDemand.cpp.

Referenced by main(), and readConfiguration().

**11.36 trademgen/batches/trademgen\_with\_db.cpp File Reference**

```
#include <cassert>
#include <iostream>
#include <sstream>
#include <fstream>
#include <vector>
```

```
#include <string>
#include <boost/date_time/posix_time/posix_time.hpp>
#include <boost/date_time/gregorian/gregorian.hpp>
#include <boost/tokenizer.hpp>
#include <boost/program_options.hpp>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/basic/BasConst_General.hpp>
#include <stdair/basic/BasDBParams.hpp>
#include <stdair/basic/BasLogParams.hpp>
#include <trademgen/TRADEMGEN_Service.hpp>
#include <trademgen/config/trademgen-paths.hpp>
```

## Typedefs

- typedef std::vector< std::string > [WordList\\_T](#)

## Functions

- const std::string [K\\_TRADEMGEN\\_DEFAULT\\_LOG\\_FILENAME](#) ("trademgen\_with\_db.log")
- const std::string [K\\_TRADEMGEN\\_DEFAULT\\_INPUT\\_FILENAME](#) (STDAIR\_SAMPLE\_DIR"/demand01.csv")
- const std::string [K\\_TRADEMGEN\\_DEFAULT\\_QUERY\\_STRING](#) ("my good old query")
- const std::string [K\\_TRADEMGEN\\_DEFAULT\\_DB\\_USER](#) ("dsim")
- const std::string [K\\_TRADEMGEN\\_DEFAULT\\_DB\\_PASSWD](#) ("dsim")
- const std::string [K\\_TRADEMGEN\\_DEFAULT\\_DB\\_DBNAME](#) ("sim\_dsim")
- const std::string [K\\_TRADEMGEN\\_DEFAULT\\_DB\\_HOST](#) ("localhost")
- const std::string [K\\_TRADEMGEN\\_DEFAULT\\_DB\\_PORT](#) ("3306")
- void [tokeniseStringIntoWordList](#) (const std::string &iPhrase, [WordList\\_T](#) &iWordList)
- std::string [createStringFromWordList](#) (const [WordList\\_T](#) &iWordList)
- template<class T> std::ostream & [operator<<](#) (std::ostream &os, const std::vector< T > &v)
- int [readConfiguration](#) (int argc, char \*argv[], bool &ioIsBuiltin, stdair::RandomSeed\_T &ioRandomSeed, std::string &ioQueryString, stdair::Filename\_T &ioInputFilename, std::string &ioLogFilename, std::string &ioDBUser, std::string &ioDBPasswd, std::string &ioDBHost, std::string &ioDBPort, std::string &ioDBDBName)
- int [main](#) (int argc, char \*argv[ ])

## Variables

- const bool [K\\_TRADEMGEN\\_DEFAULT\\_BUILT\\_IN\\_INPUT](#) = false
- const stdair::RandomSeed\_T [K\\_TRADEMGEN\\_DEFAULT\\_RANDOM\\_SEED](#)
- const int [K\\_TRADEMGEN\\_EARLY\\_RETURN\\_STATUS](#) = 99

### 11.36.1 Typedef Documentation

#### 11.36.1.1 typedef std::vector<std::string> [WordList\\_T](#)

Definition at line 24 of file trademgen\_with\_db.cpp.

### 11.36.2 Function Documentation

#### 11.36.2.1 `const std::string K_TRADEMGEN_DEFAULT_LOG_FILENAME ("trademgen_with_db.log")`

Default name and location for the log file.

#### 11.36.2.2 `const std::string K_TRADEMGEN_DEFAULT_INPUT_FILENAME (STDAIR_SAMPLE_DIR"/demand01.csv")`

Default name and location for the (CSV) input file.

#### 11.36.2.3 `const std::string K_TRADEMGEN_DEFAULT_QUERY_STRING ("my good old query")`

Default query string.

Referenced by `readConfiguration()`.

#### 11.36.2.4 `const std::string K_TRADEMGEN_DEFAULT_DB_USER ("dsim")`

Default parameters for the database connection.

Referenced by `readConfiguration()`.

#### 11.36.2.5 `const std::string K_TRADEMGEN_DEFAULT_DB_PASSWD ("dsim")`

Referenced by `readConfiguration()`.

#### 11.36.2.6 `const std::string K_TRADEMGEN_DEFAULT_DB_DBNAME ("sim_dsim")`

Referenced by `readConfiguration()`.

#### 11.36.2.7 `const std::string K_TRADEMGEN_DEFAULT_DB_HOST ("localhost")`

Referenced by `readConfiguration()`.

#### 11.36.2.8 `const std::string K_TRADEMGEN_DEFAULT_DB_PORT ("3306")`

Referenced by `readConfiguration()`.

#### 11.36.2.9 `void tokeniseStringIntoWordList (const std::string & iPhrase, WordList_T & ioWordList)`

Definition at line 67 of file `trademgen_with_db.cpp`.

Referenced by `readConfiguration()`.

#### 11.36.2.10 `std::string createStringFromWordList (const WordList_T & iWordList)`

Definition at line 89 of file `trademgen_with_db.cpp`.

Referenced by `readConfiguration()`.

**11.36.2.11** `template<class T> std::ostream& operator<< (std::ostream & os, const std::vector< T > & v)`

Definition at line 108 of file trademgen\_with\_db.cpp.

**11.36.2.12** `int readConfiguration (int argc, char * argv[ ], bool & ioIsBuiltin, stdair::RandomSeed_T & ioRandomSeed, std::string & ioQueryString, stdair::Filename_T & ioInputFilename, std::string & ioLogFilename, std::string & ioDBUser, std::string & ioDBPasswd, std::string & ioDBHost, std::string & ioDBPort, std::string & ioDBDBName)`

Read and parse the command line options.

Definition at line 118 of file trademgen\_with\_db.cpp.

References `createStringFromWordList()`, `K_TRADEMGEN_DEFAULT_BUILT_IN_INPUT`, `K_TRADEMGEN_DEFAULT_DB_DBNAME()`, `K_TRADEMGEN_DEFAULT_DB_HOST()`, `K_TRADEMGEN_DEFAULT_DB_PASSWD()`, `K_TRADEMGEN_DEFAULT_DB_PORT()`, `K_TRADEMGEN_DEFAULT_DB_USER()`, `K_TRADEMGEN_DEFAULT_INPUT_FILENAME()`, `K_TRADEMGEN_DEFAULT_LOG_FILENAME()`, `K_TRADEMGEN_DEFAULT_QUERY_STRING()`, `K_TRADEMGEN_DEFAULT_RANDOM_SEED`, `K_TRADEMGEN_EARLY_RETURN_STATUS`, and `tokeniseStringIntoWordList()`.

**11.36.2.13** `int main (int argc, char * argv[ ])`

Definition at line 288 of file trademgen\_with\_db.cpp.

References `TRADEMGEN::TRADEMGEN_Service::buildSampleBom()`, `TRADEMGEN::TRADEMGEN_Service::displayAirlineListFromDB()`, `K_TRADEMGEN_EARLY_RETURN_STATUS`, `TRADEMGEN::TRADEMGEN_Service::parseAndLoad()`, and `readConfiguration()`.

**11.36.3 Variable Documentation****11.36.3.1** `const bool K_TRADEMGEN_DEFAULT_BUILT_IN_INPUT = false`

Default for the input type. It can be either built-in or provided by an input file. That latter must then be given with the `-i` option.

Definition at line 43 of file trademgen\_with\_db.cpp.

**11.36.3.2** `const stdair::RandomSeed_T K_TRADEMGEN_DEFAULT_RANDOM_SEED`

**Initial value:**

```
stdair::DEFAULT_RANDOM_SEED
```

Default random generation seed (e.g., 120765987).

Definition at line 48 of file trademgen\_with\_db.cpp.

**11.36.3.3** `const int K_TRADEMGEN_EARLY_RETURN_STATUS = 99`

Early return status (so that it can be differentiated from an error).

Definition at line 115 of file trademgen\_with\_db.cpp.



## 11.37 trademgen/bom/BomDisplay.cpp File Reference

```
#include <cassert>
#include <ostream>
#include <stdair/basic/BasConst_BomDisplay.hpp>
#include <stdair/bom/BomManager.hpp>
#include <sevmgr/SEVMGR_Service.hpp>
#include <sevmgr/SEVMGR_Types.hpp>
#include <trademgen/bom/DemandStream.hpp>
#include <trademgen/bom/BomDisplay.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

### Classes

- struct [TRADEMGEN::FlagSaver](#)

## 11.38 trademgen/bom/BomDisplay.hpp File Reference

```
#include <iosfwd>
#include <string>
#include <sevmgr/SEVMGR_Types.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

### Classes

- class [TRADEMGEN::BomDisplay](#)  
*Utility class to display TraDemGen objects with a pretty format.*

## 11.39 trademgen/bom/DemandStream.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <cmath>
#include <iomanip>
#include <boost/make_shared.hpp>
#include <stdair/basic/BasConst_General.hpp>
```

```
#include <stdair/basic/BasConst_Inventory.hpp>
#include <stdair/basic/BasConst_Request.hpp>
#include <stdair/bom/BookingRequestStruct.hpp>
#include <stdair/service/Logger.hpp>
#include <trademgen/basic/BasConst_DemandGeneration.hpp>
#include <trademgen/bom/DemandStream.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

## 11.40 trademgen/bom/DemandStream.hpp File Reference

```
#include <stdair/bom/BomAbstract.hpp>
#include <stdair/bom/BookingRequestTypes.hpp>
#include <stdair/basic/RandomGeneration.hpp>
#include <stdair/basic/DemandGenerationMethod.hpp>
#include <trademgen/basic/DemandCharacteristics.hpp>
#include <trademgen/basic/DemandDistribution.hpp>
#include <trademgen/basic/RandomGenerationContext.hpp>
#include <trademgen/bom/DemandStreamKey.hpp>
#include <trademgen/bom/DemandStreamTypes.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [TRADEMGEN](#)

### Classes

- class [TRADEMGEN::DemandStream](#)  
*Class modeling a demand stream.*

## 11.41 trademgen/bom/DemandStreamKey.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/basic/BasConst_Inventory.hpp>
#include <trademgen/bom/DemandStreamKey.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

## 11.42 trademgen/bom/DemandStreamKey.hpp File Reference

```
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_date_time_types.hpp>
#include <stdair/bom/KeyAbstract.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

### Classes

- struct [TRADEMGEN::DemandStreamKey](#)

## 11.43 trademgen/bom/DemandStreamTypes.hpp File Reference

```
#include <map>
#include <list>
#include <stdair/bom/key_types.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

### Typedefs

- typedef std::list< DemandStream \* > [TRADEMGEN::DemandStreamList\\_T](#)
- typedef std::map< const stdair::MapKey\_T, DemandStream \* > [TRADEMGEN::DemandStream-Map\\_T](#)

## 11.44 trademgen/bom/DemandStruct.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/basic/BasConst_Inventory.hpp>
#include <stdair/basic/BasConst_Period_BOM.hpp>
#include <stdair/service/Logger.hpp>
#include <trademgen/TRADEMGEN_Types.hpp>
#include <trademgen/bom/DemandStruct.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

## 11.45 trademgen/bom/DemandStruct.hpp File Reference

```
#include <string>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_maths_types.hpp>
#include <stdair/stdair_date_time_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
#include <stdair/bom/DoWStruct.hpp>
#include <trademgen/basic/DemandCharacteristicsTypes.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

### Classes

- struct [TRADEMGEN::DemandStruct](#)

## 11.46 trademgen/command/DBManager.cpp File Reference

```
#include <cassert>
#include <soci/soci.h>
#include <soci/mysql/soci-mysql.h>
#include <stdair/bom/AirlineStruct.hpp>
#include <stdair/service/Logger.hpp>
#include <trademgen/command/DBManager.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

## 11.47 trademgen/command/DBManager.hpp File Reference

```
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_db.hpp>
#include <trademgen/TRADEMGEN_Types.hpp>
```

## Namespaces

- namespace [TRADEMGEN](#)

## Classes

- class [TRADEMGEN::DBManager](#)

## 11.48 trademgen/command/DemandManager.cpp File Reference

```
#include <cassert>
#include <boost/make_shared.hpp>
#include <stdair/basic/ProgressStatusSet.hpp>
#include <stdair/basic/BasConst_Request.hpp>
#include <stdair/bom/BomManager.hpp>
#include <stdair/bom/EventStruct.hpp>
#include <stdair/bom/BookingRequestStruct.hpp>
#include <stdair/bom/TravelSolutionStruct.hpp>
#include <stdair/bom/CancellationStruct.hpp>
#include <stdair/factory/FacBom.hpp>
#include <stdair/factory/FacBomManager.hpp>
#include <stdair/service/Logger.hpp>
#include <sevmgr/SEVMGR_Service.hpp>
#include <trademgen/basic/DemandCharacteristics.hpp>
#include <trademgen/basic/DemandDistribution.hpp>
#include <trademgen/bom/DemandStruct.hpp>
#include <trademgen/bom/DemandStream.hpp>
#include <trademgen/command/DemandManager.hpp>
```

## Namespaces

- namespace [TRADEMGEN](#)

## 11.49 trademgen/command/DemandManager.hpp File Reference

```
#include <stdair/stdair_basic_types.hpp>
#include <stdair/basic/RandomGeneration.hpp>
#include <stdair/basic/DemandGenerationMethod.hpp>
#include <stdair/bom/BookingRequestTypes.hpp>
#include <stdair/command/CmdAbstract.hpp>
```

```
#include <sevmgr/SEVMGR_Types.hpp>
#include <trademgen/TRADEMGEN_Types.hpp>
#include <trademgen/basic/DemandCharacteristicsTypes.hpp>
#include <trademgen/bom/DemandStreamKey.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [TRADEMGEN](#)
- namespace [TRADEMGEN::DemandParserHelper](#)

### Classes

- class [TRADEMGEN::DemandManager](#)  
*Utility class for Demand and [DemandStream](#) objects.*

## 11.50 trademgen/command/DemandParser.cpp File Reference

```
#include <cassert>
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/basic/RandomGeneration.hpp>
#include <stdair/bom/Inventory.hpp>
#include <trademgen/command/DemandParserHelper.hpp>
#include <trademgen/command/DemandParser.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

## 11.51 trademgen/command/DemandParser.hpp File Reference

```
#include <string>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/command/CmdAbstract.hpp>
#include <sevmgr/SEVMGR_Types.hpp>
#include <trademgen/TRADEMGEN_Types.hpp>
#include <trademgen/basic/DemandCharacteristicsTypes.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [TRADEMGEN](#)

## Classes

- class [TRADEMGEN::DemandParser](#)  
*Class wrapping the parser entry point.*

## 11.52 trademgen/command/DemandParserHelper.cpp File Reference

```
#include <cassert>
#include <stdair/basic/RandomGeneration.hpp>
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/service/Logger.hpp>
#include <trademgen/basic/DemandCharacteristicsTypes.hpp>
#include <trademgen/command/DemandParserHelper.hpp>
#include <trademgen/command/DemandManager.hpp>
```

## Namespaces

- namespace [TRADEMGEN](#)
- namespace [TRADEMGEN::DemandParserHelper](#)

## Functions

- [repeat\\_p\\_t TRADEMGEN::DemandParserHelper::airline\\_code\\_p](#) ([chset\\_t](#)("0-9A-Z").derived(), 2, 3)
- [bounded1\\_4\\_p\\_t TRADEMGEN::DemandParserHelper::flight\\_number\\_p](#) ([uint1\\_4\\_p](#).derived(), 0u, 9999u)
- [bounded4\\_p\\_t TRADEMGEN::DemandParserHelper::year\\_p](#) ([uint4\\_p](#).derived(), 2000u, 2099u)
- [bounded2\\_p\\_t TRADEMGEN::DemandParserHelper::month\\_p](#) ([uint2\\_p](#).derived(), 1u, 12u)
- [bounded2\\_p\\_t TRADEMGEN::DemandParserHelper::day\\_p](#) ([uint2\\_p](#).derived(), 1u, 31u)
- [repeat\\_p\\_t TRADEMGEN::DemandParserHelper::dow\\_p](#) ([chset\\_t](#)("0-1").derived().derived(), 7, 7)
- [repeat\\_p\\_t TRADEMGEN::DemandParserHelper::airport\\_p](#) ([chset\\_t](#)("0-9A-Z").derived(), 3, 3)
- [bounded1\\_2\\_p\\_t TRADEMGEN::DemandParserHelper::hours\\_p](#) ([uint1\\_2\\_p](#).derived(), 0u, 23u)
- [bounded2\\_p\\_t TRADEMGEN::DemandParserHelper::minutes\\_p](#) ([uint2\\_p](#).derived(), 0u, 59u)
- [bounded2\\_p\\_t TRADEMGEN::DemandParserHelper::seconds\\_p](#) ([uint2\\_p](#).derived(), 0u, 59u)
- [chset\\_t TRADEMGEN::DemandParserHelper::cabin\\_code\\_p](#) ("A-Z")
- [chset\\_t TRADEMGEN::DemandParserHelper::passenger\\_type\\_p](#) ("A-Z")
- [chset\\_t TRADEMGEN::DemandParserHelper::ff\\_type\\_p](#) ("A-Z")
- [repeat\\_p\\_t TRADEMGEN::DemandParserHelper::class\\_code\\_list\\_p](#) ([chset\\_t](#)("A-Z").derived(), 1, 26)
- [bounded1\\_3\\_p\\_t TRADEMGEN::DemandParserHelper::stay\\_duration\\_p](#) ([uint1\\_3\\_p](#).derived(), 0u, 999u)

### Variables

- [int1\\_p\\_t TRADEMGEN::DemandParserHelper::int1\\_p](#)
- [uint2\\_p\\_t TRADEMGEN::DemandParserHelper::uint2\\_p](#)
- [uint1\\_2\\_p\\_t TRADEMGEN::DemandParserHelper::uint1\\_2\\_p](#)
- [uint1\\_3\\_p\\_t TRADEMGEN::DemandParserHelper::uint1\\_3\\_p](#)
- [uint4\\_p\\_t TRADEMGEN::DemandParserHelper::uint4\\_p](#)
- [uint1\\_4\\_p\\_t TRADEMGEN::DemandParserHelper::uint1\\_4\\_p](#)
- [int1\\_p\\_t TRADEMGEN::DemandParserHelper::family\\_code\\_p](#)

## 11.53 trademgen/command/DemandParserHelper.hpp File Reference

```
#include <string>
#include <stdair/command/CmdAbstract.hpp>
#include <sevmgr/SEVMGR_Types.hpp>
#include <trademgen/TRADEMGEN_Types.hpp>
#include <trademgen/basic/BasParserTypes.hpp>
#include <trademgen/bom/DemandStruct.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [TRADEMGEN](#)
- namespace [TRADEMGEN::DemandParserHelper](#)

### Classes

- struct [TRADEMGEN::DemandParserHelper::ParserSemanticAction](#)
- struct [TRADEMGEN::DemandParserHelper::storePrefDepDateRangeStart](#)
- struct [TRADEMGEN::DemandParserHelper::storePrefDepDateRangeEnd](#)
- struct [TRADEMGEN::DemandParserHelper::storeDow](#)
- struct [TRADEMGEN::DemandParserHelper::storeOrigin](#)
- struct [TRADEMGEN::DemandParserHelper::storeDestination](#)
- struct [TRADEMGEN::DemandParserHelper::storePrefCabin](#)
- struct [TRADEMGEN::DemandParserHelper::storeDemandMean](#)
- struct [TRADEMGEN::DemandParserHelper::storeDemandStdDev](#)
- struct [TRADEMGEN::DemandParserHelper::storeDemandChangeFeeProb](#)
- struct [TRADEMGEN::DemandParserHelper::storeDemandChangeFeeDisutility](#)
- struct [TRADEMGEN::DemandParserHelper::storeDemandNonRefundableProb](#)
- struct [TRADEMGEN::DemandParserHelper::storeDemandNonRefundableDisutility](#)
- struct [TRADEMGEN::DemandParserHelper::storePosCode](#)
- struct [TRADEMGEN::DemandParserHelper::storePosProbMass](#)
- struct [TRADEMGEN::DemandParserHelper::storeChannelCode](#)
- struct [TRADEMGEN::DemandParserHelper::storeChannelProbMass](#)
- struct [TRADEMGEN::DemandParserHelper::storeTripCode](#)
- struct [TRADEMGEN::DemandParserHelper::storeTripProbMass](#)
- struct [TRADEMGEN::DemandParserHelper::storeStayCode](#)



- struct [TRADEMGEN::DemandParserHelper::storeStayProbMass](#)
- struct [TRADEMGEN::DemandParserHelper::storeFFCode](#)
- struct [TRADEMGEN::DemandParserHelper::storeFFProbMass](#)
- struct [TRADEMGEN::DemandParserHelper::storePrefDepTime](#)
- struct [TRADEMGEN::DemandParserHelper::storePrefDepTimeProbMass](#)
- struct [TRADEMGEN::DemandParserHelper::storeWTP](#)
- struct [TRADEMGEN::DemandParserHelper::storeTimeValue](#)
- struct [TRADEMGEN::DemandParserHelper::storeTimeValueProbMass](#)
- struct [TRADEMGEN::DemandParserHelper::storeDTD](#)
- struct [TRADEMGEN::DemandParserHelper::storeDTDProbMass](#)
- struct [TRADEMGEN::DemandParserHelper::doEndDemand](#)
- struct [TRADEMGEN::DemandParserHelper::DemandParser](#)
- struct [TRADEMGEN::DemandParserHelper::DemandParser::definition< ScannerT >](#)
- class [TRADEMGEN::DemandFileParser](#)

## 11.54 trademgen/DBParams.hpp File Reference

```
#include <iosfwd>
#include <string>
#include <trademgen/TRADEMGEN_Types.hpp>
#include <trademgen/TRADEMGEN_Abstract.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

### Classes

- struct [TRADEMGEN::DBParams](#)

### Typedefs

- typedef std::list< std::string > [TRADEMGEN::DBParamsNameList\\_T](#)

## 11.55 trademgen/factory/FacTRADEMGENServiceContext.cpp File Reference

```
#include <cassert>
#include <stdair/service/FacSupervisor.hpp>
#include <trademgen/factory/FacTRADEMGENServiceContext.hpp>
#include <trademgen/service/TRADEMGEN_ServiceContext.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

## 11.56 trademgen/factory/FacTRADEMGENSEerviceContext.hpp File Reference

```
#include <stdair/stdair_maths_types.hpp>
#include <stdair/service/FacServiceAbstract.hpp>
#include <trademgen/TRADEMGEN_Types.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

### Classes

- class [TRADEMGEN::FacTRADEMGENSEerviceContext](#)  
*Factory for creating the TraDemGen service context instance.*

## 11.57 trademgen/python/pytrademgen.cpp File Reference

```
#include <cassert>
#include <stdexcept>
#include <fstream>
#include <sstream>
#include <string>
#include <list>
#include <vector>
#include <boost/python.hpp>
#include <boost/accumulators/accumulators.hpp>
#include <boost/accumulators/statistics.hpp>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/basic/BasConst_General.hpp>
#include <stdair/basic/ProgressStatusSet.hpp>
#include <stdair/basic/DemandGenerationMethod.hpp>
#include <stdair/bom/EventStruct.hpp>
#include <stdair/bom/BookingRequestStruct.hpp>
#include <stdair/bom/BomDisplay.hpp>
#include <stdair/service/Logger.hpp>
#include <trademgen/TRADEMGEN_Service.hpp>
#include <trademgen/config/trademgen-paths.hpp>
```

## Namespaces

- namespace [TRADEMGEN](#)

## Classes

- struct [TRADEMGEN::Trademgener](#)

*Wrapper structure around the C++ API, so as to expose a Python API.*

## Typedefs

- typedef unsigned int [NbOfRuns\\_T](#)
- typedef `ba::accumulator_set< double, ba::stats< ba::tag::min, ba::tag::max, ba::tag::mean(ba::immediate), ba::tag::sum, ba::tag::variance > > stat\_acc\_type`

## Functions

- void [TRADEMGEN::stat\\_display](#) (std::ostream &oStream, const [stat\\_acc\\_type](#) &iStatAcc)
- [BOOST\\_PYTHON\\_MODULE](#) (libpytrademgen)

### 11.57.1 Typedef Documentation

#### 11.57.1.1 typedef unsigned int [NbOfRuns\\_T](#)

Definition at line 31 of file pytrademgen.cpp.

#### 11.57.1.2 typedef `ba::accumulator_set<double, ba::stats<ba::tag::min, ba::tag::max, ba::tag::mean (ba::immediate), ba::tag::sum, ba::tag::variance> > stat\_acc\_type`

Type definition to gather statistics.

Definition at line 40 of file pytrademgen.cpp.

### 11.57.2 Function Documentation

#### 11.57.2.1 [BOOST\\_PYTHON\\_MODULE](#) (libpytrademgen)

Definition at line 355 of file pytrademgen.cpp.

References [TRADEMGEN::Trademgener::init\(\)](#), and [TRADEMGEN::Trademgener::trademgen\(\)](#).

## 11.58 trademgen/service/TRADEMGEN\_Service.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <boost/make_shared.hpp>
#include <soci/soci.h>
#include <stdair/basic/BasChronometer.hpp>
```

```

#include <stdair/basic/BasConst_General.hpp>
#include <stdair/basic/ProgressStatusSet.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/bom/BookingRequestStruct.hpp>
#include <stdair/bom/AirlineStruct.hpp>
#include <stdair/bom/EventStruct.hpp>
#include <stdair/command/DBManagerForAirlines.hpp>
#include <stdair/service/Logger.hpp>
#include <stdair/service/DBSessionManager.hpp>
#include <stdair/STDAIR_Service.hpp>
#include <stdair/factory/FacBomManager.hpp>
#include <sevmgr/SEVMGR_Service.hpp>
#include <trademgen/basic/BasConst_TRADEMGEN_Service.hpp>
#include <trademgen/bom/BomDisplay.hpp>
#include <trademgen/bom/DemandStream.hpp>
#include <trademgen/bom/DemandStreamTypes.hpp>
#include <trademgen/factory/FacTRADEMGENServiceContext.hpp>
#include <trademgen/command/DemandParser.hpp>
#include <trademgen/command/DemandManager.hpp>
#include <trademgen/service/TRADEMGEN_ServiceContext.hpp>
#include <trademgen/TRADEMGEN_Service.hpp>

```

## Namespaces

- namespace [TRADEMGEN](#)
- namespace [SEVMGR](#)

## Functions

- template void [SEVMGR::SEVMGR\\_Service::addEventGenerator< TRADEMGEN::DemandStream >](#) ([TRADEMGEN::DemandStream](#) &) const
- template [TRADEMGEN::DemandStream](#) & [SEVMGR::SEVMGR\\_Service::getEventGenerator< TRADEMGEN::DemandStream, stdair::DemandStreamKeyStr\\_T >](#) (const stdair::DemandStreamKeyStr\_T &) const
- template bool [SEVMGR::SEVMGR\\_Service::hasEventGenerator< TRADEMGEN::DemandStream, stdair::DemandStreamKeyStr\\_T >](#) (const stdair::DemandStreamKeyStr\_T &) const
- template const [TRADEMGEN::DemandStreamList\\_T](#) [SEVMGR::SEVMGR\\_Service::getEventGeneratorList< TRADEMGEN::DemandStream >](#) () const
- template bool [SEVMGR::SEVMGR\\_Service::hasEventGeneratorList< TRADEMGEN::DemandStream >](#) () const

## 11.59 trademgen/service/TRADEMGEN\_ServiceContext.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/STDAIR_Service.hpp>
#include <stdair/basic/BasConst_General.hpp>
#include <trademgen/basic/BasConst_DemandGeneration.hpp>
#include <trademgen/service/TRADEMGEN_ServiceContext.hpp>
```

### Namespaces

- namespace [TRADEMGEN](#)

## 11.60 trademgen/service/TRADEMGEN\_ServiceContext.hpp File Reference

```
#include <string>
#include <boost/shared_ptr.hpp>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_service_types.hpp>
#include <stdair/basic/RandomGeneration.hpp>
#include <stdair/bom/BookingRequestTypes.hpp>
#include <stdair/service/ServiceAbstract.hpp>
#include <sevmgr/SEVMGR_Types.hpp>
#include <trademgen/TRADEMGEN_Types.hpp>
#include <trademgen/basic/DemandCharacteristicsTypes.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [TRADEMGEN](#)

### Classes

- class [TRADEMGEN::TRADEMGEN\\_ServiceContext](#)  
*Class holding the context of the Trademgen services.*

## 11.61 trademgen/TRADEMGEN\_Abstract.hpp File Reference

```
#include <istream>
#include <ostream>
#include <sstream>
```

```
#include <string>
```

## Namespaces

- namespace [TRADEMGEN](#)

## Classes

- struct [TRADEMGEN::TRADEMGEN\\_Abstract](#)

## Functions

- `template<class charT, class traits> std::basic_ostream< charT, traits > & operator<< (std::basic_ostream< charT, traits > &ioOut, const TRADEMGEN::TRADEMGEN\_Abstract &iStructure)`
- `template<class charT, class traits> std::basic_istream< charT, traits > & operator>> (std::basic_istream< charT, traits > &ioIn, TRADEMGEN::TRADEMGEN\_Abstract &ioStructure)`

### 11.61.1 Function Documentation

**11.61.1.1** `template<class charT, class traits> std::basic_ostream<charT, traits>& operator<< (std::basic_ostream< charT, traits > & ioOut, const TRADEMGEN::TRADEMGEN\_Abstract & iStructure) [inline]`

Piece of code given by Nicolai M. Josuttis, Section 13.12.1 "Implementing Output Operators" (p653) of his book "The C++ Standard Library: A Tutorial and Reference", published by Addison-Wesley.

Definition at line 49 of file TRADEMGEN\_Abstract.hpp.

**11.61.1.2** `template<class charT, class traits> std::basic_istream<charT, traits>& operator>> (std::basic_istream< charT, traits > & ioIn, TRADEMGEN::TRADEMGEN\_Abstract & ioStructure) [inline]`

Piece of code given by Nicolai M. Josuttis, Section 13.12.1 "Implementing Output Operators" (pp655-657) of his book "The C++ Standard Library: A Tutorial and Reference", published by Addison-Wesley.

Definition at line 77 of file TRADEMGEN\_Abstract.hpp.

References [TRADEMGEN::TRADEMGEN\\_Abstract::fromStream\(\)](#).

## 11.62 trademgen/TRADEMGEN\_Exceptions.hpp File Reference

```
#include <exception>
#include <stdair/stdair_exceptions.hpp>
```

## Namespaces

- namespace [TRADEMGEN](#)

## Classes

- class [TRADEMGEN::TrademgenGenerationException](#)
- class [TRADEMGEN::DemandInputFileNotFoundException](#)
- class [TRADEMGEN::IndexOutOfRangeException](#)

## 11.63 trademgen/TRADEMGEN\_Service.hpp File Reference

```
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_demand_types.hpp>
#include <stdair/stdair_maths_types.hpp>
#include <stdair/stdair_json.hpp>
#include <stdair/stdair_service_types.hpp>
#include <stdair/basic/DemandGenerationMethod.hpp>
#include <stdair/bom/BookingRequestTypes.hpp>
#include <stdair/bom/EventTypes.hpp>
#include <stdair/bom/EventStruct.hpp>
#include <sevmgr/SEVMGR_Types.hpp>
#include <trademgen/TRADEMGEN_Types.hpp>
```

## Namespaces

- namespace [stdair](#)
- namespace [TRADEMGEN](#)

## Classes

- class [TRADEMGEN::TRADEMGEN\\_Service](#)  
*class holding the services related to Travel Demand Generation.*

## 11.64 trademgen/TRADEMGEN\_Types.hpp File Reference

```
#include <boost/shared_ptr.hpp>
#include <stdair/stdair_file.hpp>
#include <trademgen/TRADEMGEN_Exceptions.hpp>
```

## Namespaces

- namespace [TRADEMGEN](#)

## Classes

- class [TRADEMGEN::DemandFilePath](#)

## Typedefs

- typedef boost::shared\_ptr< TRADEMGEN\_Service > [TRADEMGEN::TRADEMGEN\\_Service-Ptr\\_T](#)

## 11.65 trademgen/ui/cmdline/trademgen.cpp File Reference

# 12 TraDemGen Page Documentation

## 12.1 People

### 12.1.1 Project Admins (and Developers)

- Anh Quan Nguyen <quannaus@users.sourceforge.net> ([N](#))
- Denis Arnaud <denis\_arnaud@users.sourceforge.net> ([N](#))
- Gabrielle Sabatier <gsabatier@users.sourceforge.net> ([N](#))

### 12.1.2 Retired Developers

- Mehdi Ayouni <mehdi.ayouni@gmail.com>
- Son Nguyen Kim <snguyenkim@users.sourceforge.net> ([N](#))

### 12.1.3 Contributors

- Emmanuel Bastien <ebastien@users.sourceforge.net> ([N](#))

### 12.1.4 Distribution Maintainers

- [Fedora/RedHat](#): Denis Arnaud <denis\_arnaud@users.sourceforge.net> ([N](#))
- [Debian](#): Emmanuel Bastien <ebastien@users.sourceforge.net> ([N](#))

## Note:

(N) - [Amadeus](#) employees.

## 12.2 Coding Rules

In the following sections we describe the naming conventions which are used for files, classes, structures, local variables, and global variables.

### 12.2.1 Default Naming Rules for Variables

Variables names follow Java naming conventions. Examples:

- `lNumberOfPassengers`
- `lSeatAvailability`



### 12.2.2 Default Naming Rules for Functions

Function names follow Java naming conventions. Example:

- `int myFunctionName (const int& a, int b)`

### 12.2.3 Default Naming Rules for Classes and Structures

Each new word in a class or structure name should always start with a capital letter and the words should be separated with an under-score. Abbreviations are written with capital letters. Examples:

- `MyClassName`
- `MyStructName`

### 12.2.4 Default Naming Rules for Files

Files are named after the C++ class names.

Source files are named using `.cpp` suffix, whereas header files end with `.hpp` extension. Examples:

- `FlightDate.hpp`
- `SegmentDate.cpp`

### 12.2.5 Default Functionality of Classes

All classes that are configured by input parameters should include:

- default empty constructor
- one or more additional constructor(s) that takes input parameters and initializes the class instance
- setup function, preferably named `'setup'` or `'set_parameters'`

Explicit destructor functions are not required, unless they are needed. It shall not be possible to use any of the other member functions unless the class has been properly initiated with the input parameters.

## 12.3 Copyright and License

### 12.3.1 GNU LESSER GENERAL PUBLIC LICENSE

#### 12.3.1.1 Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2, hence  
the version number 2.1.]

### 12.3.2 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages—typically libraries—of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the

Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

### 12.3.3 TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small

macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library,

and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted

by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

**12.3.3.1 NO WARRANTY** 15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

### 12.3.3.2 END OF TERMS AND CONDITIONS

### 12.3.4 How to Apply These Terms to Your New Programs

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
```

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!

Source

## 12.4 Documentation Rules

### 12.4.1 General Rules

All classes in TraDemGen should be properly documented with Doxygen comments in include (.hpp) files. Source (.cpp) files should be documented according to a normal standard for well documented C++ code.

An example of how the interface of a class shall be documented in TraDemGen is shown here:

```

/ *!
 * \brief Brief description of MyClass here
 *
 * Detailed description of MyClass here. With example code if needed.
 */
class MyClass {
public:
    ///! Default constructor
    MyClass(void) { setup_done = false; }

    / *!
     * \brief Constructor that initializes the class with parameters
     *
     * Detailed description of the constructor here if needed
     *
     * \param[in] param1 Description of \a param1 here
     * \param[in] param2 Description of \a param2 here
     */
    MyClass(TYPE1 param1, TYPE2 param2) { setup(param1, param2); }

    / *!
     * \brief Setup function for MyClass
     *
     * Detailed description of the setup function here if needed
     *
     * \param[in] param1 Description of \a param1 here
     * \param[in] param2 Description of \a param2 here
     */
    void setup(TYPE1 param1, TYPE2 param2);

    / *!
     * \brief Brief description of memberFunction1
     *
     * Detailed description of memberFunction1 here if needed
     *
     * \param[in] param1 Description of \a param1 here
     * \param[in] param2 Description of \a param2 here
     * \param[in,out] param3 Description of \a param3 here
     * \return Description of the return value here
     */
    TYPE4 memberFunction1(TYPE1 param1, TYPE2 param2, TYPE3 &param3);

private:

```



```

bool _setupDone;          /*!< Variable that checks if the class is properly
                           initialized with parameters */
TYPE1 _privateVariable1; /*!< Short description of _privateVariable1 here
TYPE2 _privateVariable2; /*!< Short description of _privateVariable2 here
};

```

### 12.4.2 File Header

All files should start with the following header, which include Doxygen's `\file`, `\brief` and `\author` tags, `$Date$` and `$Revisions$` CVS tags, and a common copyright note:

```

/*!
 * \file
 * \brief Brief description of the file here
 * \author Names of the authors who contributed to this code
 * \date Date
 *
 * Detailed description of the file here if needed.
 *
 * -----
 *
 * TraDemGen - C++ Simulated Revenue Accounting (RAC) System Library
 *
 * Copyright (C) 2009-2011 (\see authors file for a list of contributors)
 *
 * \see copyright file for license information
 *
 * -----
 */

```

### 12.4.3 Grouping Various Parts

All functions must be added to a Doxygen group in order to appear in the documentation. The following code example defines the group `'my_group'`:

```

/*!
 * \defgroup my_group Brief description of the group here
 *
 * Detailed description of the group here
 */

```

The following example shows how to document the function `myFunction` and how to add it to the group `my_group`:

```

/*!
 * \brief Brief description of myFunction here
 * \ingroup my_group
 *
 * Detailed description of myFunction here
 *
 * \param[in] param1 Description of \a param1 here
 * \param[in] param2 Description of \a param2 here
 * \return Description of the return value here
 */
TYPE3 myFunction(TYPE1 param1, TYPE2 &param2);

```

## 12.5 Main features

A short list of the main features of TraDemGen is given below sorted in different categories. Many more features and functions exist and for these we refer to the reference documentation.

### 12.5.1 Demand generation

The demand can be generated thanks to two relatively advanced pieces of algorithm, both following a sequential principle. That is, the events (booking requests) are generated one after the other, sequentially, rather than being generated all at once at the beginning of the process (e.g., a simulation).

The two sequential methods are:

- 'Intuitive' method. The booking period is sliced in `intervals`, where the arrival rate of events (booking requests) is known for each of those intervals, say  $\lambda_i$ . The inter-arrival process then follows an `exponential law`. That is, the final number of booking requests follows a `Non homogeneous Poisson distribution`. With that method, the `variance` of that `distribution` is therefore equal to the `mean`.
- 'Advanced' method. The process uses `order statistics` in order to mimic the behaviour of `uniform distributions` projected onto the known arrival pattern of events. With that method, the final number of booking requests is first drawn, following any probability distribution (e.g., `normal`, `Gamma`, `Beta` or even Weibull law) with any required standard deviation. Then, each booking request is drawn in sequence:
  - according to a mere `uniform distribution`,
  - and projected onto the known booking arrival pattern.

### 12.5.2 Other features

- CSV input file parsing
- Memory handling

## 12.6 Make a Difference

**Do not ask what TraDemGen can do for you. Ask what you can do for TraDemGen.**

You can help us to develop the TraDemGen library. There are always a lot of things you can do:

- Start using TraDemGen
- Tell your friends about TraDemGen and help them to get started using it
- If you find a bug, report it to us. Without your help we can never hope to produce a bug free code.
- Help us to improve the documentation by providing information about documentation bugs
- Answer support requests in the TraDemGen discussion forums on SourceForge. If you know the answer to a question, help others to overcome their TraDemGen problems.
- Help us to improve our algorithms. If you know of a better way (e.g. that is faster or requires less memory) to implement some of our algorithms, then let us know.

- Help us to port TraDemGen to new platforms. If you manage to compile TraDemGen on a new platform, then tell us how you did it.
- Send us your code. If you have a good TraDemGen compatible code, which you can release under the LGPL, and you think it should be included in TraDemGen, then send it to us.
- Become an TraDemGen developer. Send us an e-mail and tell what you can do for TraDemGen.

## 12.7 Make a new release

### 12.7.1 Introduction

This document describes briefly the recommended procedure of releasing a new version of TraDemGen using a Linux development machine and the SourceForge project site.

The following steps are required to make a release of the distribution package.

### 12.7.2 Initialisation

Clone locally the full [Git project](#):

```
cd ~
mkdir -p dev/sim
cd ~/dev/sim
git clone git://trademgen.git.sourceforge.net/gitroot/trademgen/trademgen trademgengit
cd trademgengit
git checkout trunk
```

### 12.7.3 Release branch maintenance

Switch to the release branch, on your local clone, and merge the latest updates from the trunk. Decide about the new version to be released.

```
cd ~/dev/sim/trademgengit
git checkout releases
git merge trunk
```

Update the version in the various build system files, replacing the old version numbers by the correct ones:

```
vi CMakeLists.txt
vi autogen.sh
vi README
```

Update the version, add some news in the NEWS file, add a change-log in the ChangeLog file and in the RPM specification files:

```
vi NEWS
vi ChangeLog
vi trademgen.spec
```

### 12.7.4 Commit and publish the release branch

Commit the new release:

```
cd ~/dev/sim/trademgengit
git add -A
git commit -m "[Release 0.5.0] Release of the 0.5.0 version of TraDemGen."
git push
```

### 12.7.5 Create distribution packages

Create the distribution packages using the following command:

```
cd ~/dev/sim/trademgengit
git checkout releases
rm -rf build && mkdir -p build
cd build
export INSTALL_BASEDIR=/home/user/dev/deliveries
export LIBSUFFIX_4_CMAKE="-DLIB_SUFFIX=64"
cmake -DCMAKE_INSTALL_PREFIX=${INSTALL_BASEDIR}/trademgen-0.5.0 \
-DWITH_STDAIR_PREFIX=${INSTALL_BASEDIR}/stdair-stable \
-DWITH_AIRAC_PREFIX=${INSTALL_BASEDIR}/airsched-stable \
-DWITH_AIRAC_PREFIX=${INSTALL_BASEDIR}/airrac-stable \
-DWITH_RMOL_PREFIX=${INSTALL_BASEDIR}/rmol-stable \
-DWITH_RMOL_PREFIX=${INSTALL_BASEDIR}/airinv-stable \
-DWITH_RMOL_PREFIX=${INSTALL_BASEDIR}/simfqt-stable \
-DCMAKE_BUILD_TYPE:String=Debug -DINSTALL_DOC:BOOL=ON \
${LIBSUFFIX_4_CMAKE} ..
make check && make dist
make install
```

This will configure, compile and check the package. The output packages will be named, for instance, `trademgen-0.5.0.tar.gz` and `trademgen-0.5.0.tar.bz2`.

### 12.7.6 Upload the HTML documentation to SourceForge

In order to update the Web site files, either:

- **synchronise them with rsync and SSH:** Upload the just generated HTML (and PDF) documentation onto the **SourceForge Web site**.

```
cd ~/dev/sim/trademgengit/build
git checkout releases
rsync -aiv ${INSTALL_BASEDIR}/trademgen-0.5.0/share/doc/trademgen-0.5.0/html/ \
your_sf_user,trademgen@web.sourceforge.net:htdocs/
```

where `-aiv` options mean:

- `-a`: archive/mirror mode; equals `-rlptgoD` (no `-H`, `-A`, `-X`)
- `-v`: increase verbosity
- `-i`: output a change-summary for all updates
- Note the trailing slashes (/) at the end of both the source and target directories. It means that the content of the source directory (`doc/html`), rather than the directory itself, has to be copied into the content of the target directory.
- or use the **SourceForge Shell service**.

### 12.7.7 Generate the RPM packages

Optionally, generate the RPM package (for instance, for [Fedora/RedHat](#)):

```
cd ~/dev/sim/trademgengit/build
git checkout releases
make dist
```

To perform this step, rpm-build, rpmlint and rpmdevtools have to be available on the system.

```
cp ../trademgen.spec ~/dev/packages/SPECS \
  && cp trademgen-0.5.0.tar.bz2 ~/dev/packages/SOURCES
cd ~/dev/packages/SPECS
rpmbuild -ba trademgen.spec
cd ~/dev/packages
rpmlint -i SPECS/trademgen.spec SRPMS/trademgen-0.5.0-1.fc16.src.rpm \
  RPMS/noarch/trademgen-* RPMS/i686/trademgen-*
```

### 12.7.8 Update distributed change log

Update the NEWS and ChangeLog files with appropriate information, including what has changed since the previous release. Then commit and push the changes into the [TraDemGen's Git repository](#).

### 12.7.9 Create the binary package, including the documentation

Create the binary package, which includes HTML and PDF documentation, using the following command:

```
cd ~/dev/sim/trademgengit/build
git checkout releases
make package
```

The output binary package will be named, for instance, trademgen-0.5.0-Linux.tar.bz2. That package contains both the HTML and PDF documentation. The binary package contains also the executables and shared libraries, as well as C++ header files, but all of those do not interest us for now.

### 12.7.10 Upload the files to SourceForge

Upload the distribution and documentation packages to the SourceForge server. Check [SourceForge help page on uploading software](#).

### 12.7.11 Make a new post

- submit a new entry in the [SourceForge project-related news feed](#)
- make a new post on the [SourceForge hosted WordPress blog](#)
- and update, if necessary, [Trac tickets](#).

### 12.7.12 Send an email on the announcement mailing-list

Finally, you should send an announcement to [trademgen-announce@lists.sourceforge.net](mailto:trademgen-announce@lists.sourceforge.net) (see <https://lists.sourceforge.net/lists/listinfo/trademgen-announce> for the archives)

## 12.8 Installation

### 12.8.1 Table of Contents

- [Fedora/RedHat Linux distributions](#)
- [TraDemGen Requirements](#)
- [Basic Installation](#)
- [Compilers and Options](#)
- [Compiling For Multiple Architectures](#)
- [Installation Names](#)
- [Optional Features](#)
- [Particular systems](#)
- [Specifying the System Type](#)
- [Sharing Defaults](#)
- [Defining Variables](#)
- [‘cmake’ Invocation](#)

### 12.8.2 Fedora/RedHat Linux distributions

Note that on [Fedora/RedHat](#) Linux distributions, RPM packages are available and can be installed with your usual package manager. For instance:

```
yum -y install trademgen-devel trademgen-doc
```

RPM packages can also be available on the [SourceForge download site](#).

### 12.8.3 TraDemGen Requirements

TraDemGen should compile without errors or warnings on most GNU/Linux systems, on UNIX systems like Solaris SunOS, and on POSIX based environments for Microsoft Windows like Cygwin or MinGW with MSYS. It can be also built on Microsoft Windows NT/2000/XP/Vista/7 using Microsoft's Visual C++ .NET, but our support for this compiler is limited. For GNU/Linux, SunOS, Cygwin and MinGW we assume that you have at least the following GNU software installed on your computer:

- GNU Autotools:
  - [autoconf](#),
  - [automake](#),
  - [libtool](#),
  - [make](#), version 3.72.1 or later (check version with `'make --version'`)
- [GCC](#) - GNU C++ Compiler (g++), version 4.3.x or later (check version with `'gcc --version'`)
- [Boost](#) - C++ STL extensions, version 1.35 or later (check version with `'grep "define BOOST_LIB_VERSION" /usr/include/boost/version.hpp'`)

- **MySQL** - Database client libraries, version 5.0 or later (check version with ``mysql --version``)
- **SOCI** - C++ database client library wrapper, version 3.0.0 or later (check version with ``soci-config-version``)

Optionally, you might need a few additional programs: **Doxygen**, **LaTeX**, **Dvips** and **Ghostscript**, to generate the HTML and PDF documentation.

We strongly recommend that you use recent stable releases of the GCC, if possible. We do not actively work on supporting older versions of the GCC, and they may therefore (without prior notice) become unsupported in future releases of TraDemGen.

#### 12.8.4 Basic Installation

Briefly, the shell commands ``./cmake .. && make install`` should configure, build and install this package. The following more-detailed instructions are generic; see the ``README`` file for instructions specific to this package. Some packages provide this ``INSTALL`` file but do not implement all of the features documented below. The lack of an optional feature in a given package is not necessarily a bug. More recommendations for GNU packages can be found in the info page corresponding to "Makefile Conventions: (standards)Makefile Conventions".

The ``cmake`` shell script attempts to guess correct values for various system-dependent variables used during compilation. It uses those values to create a ``Makefile`` in each directory of the package. It may also create one or more ``.h`` files containing system-dependent definitions. Finally, it creates a ``CMakeCache.txt`` cache file that you can refer to in the future to recreate the current configuration, and files ``CMakeFiles`` containing compiler output (useful mainly for debugging ``cmake``).

It can also use an optional file (typically called ``config.cache`` and enabled with ``-cache-file=config.cache`` or simply ``-C``) that saves the results of its tests to speed up reconfiguring. Caching is disabled by default to prevent problems with accidental use of stale cache files.

If you need to do unusual things to compile the package, please try to figure out how ``configure`` could check whether to do them, and mail diffs or instructions to the address given in the ``README`` so they can be considered for the next release. If you are using the cache, and at some point ``config.cache`` contains results you don't want to keep, you may remove or edit it.

The file ``CMakeLists.txt`` is used to create the ``Makefile`` files.

The simplest way to compile this package is:

1. ``cd`` to the directory containing the package's source code and type ``./cmake ..`` to configure the package for your system. Running ``cmake`` is generally fast. While running, it prints some messages telling which features it is checking for.
2. Type ``make`` to compile the package.
3. Optionally, type ``make check`` to run any self-tests that come with the package, generally using the just-built uninstalled binaries.
4. Type ``make install`` to install the programs and any data files and documentation. When installing into a prefix owned by root, it is recommended that the package be configured and built as a regular user, and only the ``make install`` phase executed with root privileges.
5. You can remove the program binaries and object files from the source code directory by typing ``make clean``. To also remove the files that ``configure`` created (so you can compile the package for

a different kind of computer), type `'make distclean'`. There is also a `'make maintainer-clean'` target, but that is intended mainly for the package's developers. If you use it, you may have to get all sorts of other programs in order to regenerate files that came with the distribution.

6. Often, you can also type `'make uninstall'` to remove the installed files again. In practice, not all packages have tested that uninstallation works correctly, even though it is required by the GNU Coding Standards.

### 12.8.5 Compilers and Options

Some systems require unusual options for compilation or linking that the `'cmake'` script does not know about. Run  `'./cmake -help'` for details on some of the pertinent environment variables.

You can give `'cmake'` initial values for configuration parameters by setting variables in the command line or in the environment. Here is an example:

```
./cmake CC=c99 CFLAGS=-g LIBS=-lposix
```

**See also:**

[Defining Variables](#) for more details.

### 12.8.6 Compiling For Multiple Architectures

You can compile the package for more than one kind of computer at the same time, by placing the object files for each architecture in their own directory. To do this, you can use GNU `'make'`. `'cd'` to the directory where you want the object files and executables to go and run the `'configure'` script. `'configure'` automatically checks for the source code in the directory that `'configure'` is in and in `'..'`. This is known as a `"VPATH"` build.

With a non-GNU `'make'`, it is safer to compile the package for one architecture at a time in the source code directory. After you have installed the package for one architecture, use `'make distclean'` before reconfiguring for another architecture.

On MacOS X 10.5 and later systems, you can create libraries and executables that work on multiple system types-known as `"fat"` or `"universal"` binaries-by specifying multiple `'-arch'` options to the compiler but only a single `'-arch'` option to the preprocessor. Like this:

```
./configure CC="gcc -arch i386 -arch x86_64 -arch ppc -arch ppc64" \  
           CXX="g++ -arch i386 -arch x86_64 -arch ppc -arch ppc64" \  
           CPP="gcc -E" CXXCPP="g++ -E"
```

This is not guaranteed to produce working output in all cases, you may have to build one architecture at a time and combine the results using the `'lipo'` tool if you have problems.



### 12.8.7 Installation Names

By default, `'make install'` installs the package's commands under `'/usr/local/bin'`, include files under `'/usr/local/include'`, etc. You can specify an installation prefix other than `'/usr/local'` by giving `'configure'` the option `'-prefix=PREFIX'`, where `PREFIX` must be an absolute file name.

You can specify separate installation prefixes for architecture-specific files and architecture-independent files. If you pass the option `'-exec-prefix=PREFIX'` to `'configure'`, the package uses `PREFIX` as the prefix for installing programs and libraries. Documentation and other data files still use the regular prefix.

In addition, if you use an unusual directory layout you can give options like `'-bindir=DIR'` to specify different values for particular kinds of files. Run `'configure -help'` for a list of the directories you can set and what kinds of files go in them. In general, the default for these options is expressed in terms of `'${prefix}'`, so that specifying just `'-prefix'` will affect all of the other directory specifications that were not explicitly provided.

The most portable way to affect installation locations is to pass the correct locations to `'configure'`; however, many packages provide one or both of the following shortcuts of passing variable assignments to the `'make install'` command line to change installation locations without having to reconfigure or recompile.

The first method involves providing an override variable for each affected directory. For example, `'make install prefix=/alternate/directory'` will choose an alternate location for all directory configuration variables that were expressed in terms of `'${prefix}'`. Any directories that were specified during `'configure'`, but not in terms of `'${prefix}'`, must each be overridden at install time for the entire installation to be relocated. The approach of makefile variable overrides for each directory variable is required by the GNU Coding Standards, and ideally causes no recompilation. However, some platforms have known limitations with the semantics of shared libraries that end up requiring recompilation when using this method, particularly noticeable in packages that use GNU Libtool.

The second method involves providing the `'DESTDIR'` variable. For example, `'make install DESTDIR=/alternate/directory'` will prepend `'/alternate/directory'` before all installation names. The approach of `'DESTDIR'` overrides is not required by the GNU Coding Standards, and does not work on platforms that have drive letters. On the other hand, it does better at avoiding recompilation issues, and works well even when some directory options were not specified in terms of `'${prefix}'` at `'configure'` time.

### 12.8.8 Optional Features

If the package supports it, you can cause programs to be installed with an extra prefix or suffix on their names by giving `'cmake'` the option `'-program-prefix=PREFIX'` or `'-program-suffix=SUFFIX'`.

Some packages pay attention to `'-enable-FEATURE'` options to `'configure'`, where `FEATURE` indicates an optional part of the package. They may also pay attention to `'-with-PACKAGE'` options, where `PACKAGE` is something like `'gnu-as'` or `'x'` (for the X Window System). The `'README'` should mention any `'-enable-'` and `'-with-'` options that the package recognizes.

For packages that use the X Window System, `'configure'` can usually find the X include and library files automatically, but if it doesn't, you can use the `'configure'` options `'-x-includes=DIR'` and `'-x-libraries=DIR'` to specify their locations.

Some packages offer the ability to configure how verbose the execution of `'make'` will be. For these packages, running `./configure --enable-silent-rules` sets the default to minimal output, which can be overridden with `'make V=1'`; while running `./configure --disable-silent-rules` sets the default to verbose, which can be overridden with `'make V=0'`.

### 12.8.9 Particular systems

On HP-UX, the default C compiler is not ANSI C compatible. If GNU CC is not installed, it is recommended to use the following options in order to use an ANSI C compiler:

```
./configure CC="cc -Ae -D_XOPEN_SOURCE=500"
```

and if that doesn't work, install pre-built binaries of GCC for HP-UX.

On OSF/1 a.k.a. Tru64, some versions of the default C compiler cannot parse its `<wchar.h>` header file. The option `'-nodtk'` can be used as a workaround. If GNU CC is not installed, it is therefore recommended to try

```
./configure CC="cc"
```

and if that doesn't work, try

```
./configure CC="cc -nodtk"
```

On Solaris, don't put `'/usr/ucb'` early in your `'PATH'`. This directory contains several dysfunctional programs; working variants of these programs are available in `'/usr/bin'`. So, if you need `'/usr/ucb'` in your `'PATH'`, put it `_after_` `'/usr/bin'`.

On Haiku, software installed for all users goes in `'/boot/common'`, not `'/usr/local'`. It is recommended to use the following options:

```
./cmake -DCMAKE_INSTALL_PREFIX=/boot/common
```

### 12.8.10 Specifying the System Type

There may be some features `'configure'` cannot figure out automatically, but needs to determine by the type of machine the

package will run on. Usually, assuming the package is built to be run on the `_same_` architectures, `'configure'` can figure that out, but if it prints a message saying it cannot guess the machine type, give it the `'-build=TYPE'` option. TYPE can either be a short name for the system type, such as `'sun4'`, or a canonical name which has the form CPU-COMPANY-SYSTEM

where SYSTEM can have one of these forms:

- OS
- KERNEL-OS

See the file `'config.sub'` for the possible values of each field. If `'config.sub'` isn't included in this package, then this package doesn't need to know the machine type.

If you are `_building_` compiler tools for cross-compiling, you should use the option `'-target=TYPE'` to select the type of system they will produce code for.

If you want to `_use_` a cross compiler, that generates code for a platform different from the build platform, you should specify the "host" platform (i.e., that on which the generated programs will eventually be run) with `'-host=TYPE'`.

### 12.8.11 Sharing Defaults

If you want to set default values for `'configure'` scripts to share, you can create a site shell script called `'config.site'` that gives default values for variables like `'CC'`, `'cache_file'`, and `'prefix'`. `'configure'` looks for `'PREFIX/share/config.site'` if it exists, then `'PREFIX/etc/config.site'` if it exists. Or, you can set the `'CONFIG_SITE'` environment variable to the location of the site script. A warning: not all `'configure'` scripts look for a site script.

### 12.8.12 Defining Variables

Variables not defined in a site shell script can be set in the environment passed to `'configure'`. However, some packages may run `configure` again during the build, and the customized values of these variables may be lost. In order to avoid this problem, you should set them in the `'configure'` command line, using `'VAR=value'`. For example:

```
./configure CC=/usr/local2/bin/gcc
```

causes the specified `'gcc'` to be used as the C compiler (unless it is overridden in the site shell script).

Unfortunately, this technique does not work for `'CONFIG_SHELL'` due to an Autoconf bug. Until the bug is fixed you can use this workaround:

```
CONFIG_SHELL=/bin/bash /bin/bash ./configure CONFIG_SHELL=/bin/bash
```

### 12.8.13 'cmake' Invocation

'cmake' recognizes the following options to control how it operates.

- '-help', '-h' print a summary of all of the options to 'configure', and exit.
- '-help=short', '-help=recursive' print a summary of the options unique to this package's 'configure', and exit. The 'short' variant lists options used only in the top level, while the 'recursive' variant lists options also present in any nested packages.
- '-version', '-V' print the version of Autoconf used to generate the 'configure' script, and exit.
- '-cache-file=FILE' enable the cache: use and save the results of the tests in FILE, traditionally 'config.cache'. FILE defaults to '/dev/null' to disable caching.
- '-config-cache', '-C' alias for '-cache-file=config.cache'.
- '-quiet', '-silent', '-q' do not print messages saying which checks are being made. To suppress all normal output, redirect it to '/dev/null' (any error messages will still be shown).
- '-srcdir=DIR' look for the package's source code in directory DIR. Usually 'configure' can determine that directory automatically.
- '-prefix=DIR' use DIR as the installation prefix.

**See also:**

[Installation Names](#) for more details, including other options available for fine-tuning the installation locations.

- '-no-create', '-n' run the configure checks, but stop before creating any output files.

'cmake' also accepts some other, not widely useful, options. Run 'cmake -help' for more details.

The 'cmake' script produces an output like this:

```
cmake -DCMAKE_INSTALL_PREFIX=/home/user/dev/deliveries/trademgen-99.99.99 -DLIB_SUFFIX=64 -DCMAKE_BUILD_TYPE=Release
-- The C compiler identification is GNU
-- The CXX compiler identification is GNU
-- Check for working C compiler: /usr/lib64/ccache/gcc
-- Check for working C compiler: /usr/lib64/ccache/gcc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working CXX compiler: /usr/lib64/ccache/c++
-- Check for working CXX compiler: /usr/lib64/ccache/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Requires Git without specifying any version
-- Current Git revision name: 4856624ea4978b3b2bfe26cb6702cce0be531084 trunk
-- Requires PythonLibs-2.6
-- Found PythonLibs: /usr/lib64/libpython2.7.so (Required is at least version "2.6")
-- Found PythonLibs 2.7
-- Requires Boost-1.41
```

```

-- Boost version: 1.46.0
-- Found the following Boost libraries:
--   program_options
--   date_time
--   iostreams
--   serialization
--   filesystem
--   unit_test_framework
--   python
-- Found Boost version: 1.46.0
-- Found BoostWrapper: /usr/include (Required is at least version "1.41")
-- Requires MySQL without specifying any version
-- Using mysql-config: /usr/bin/mysql_config
-- Found MySQL: /usr/lib64/mysql/libmysqlclient.so
-- Found MySQL version: 5.5.14
-- Requires SOCI-3.0
-- Using soci-config: /usr/bin/soci-config
-- SOCI headers are buried
-- Found SOCI: /usr/lib64/libsoci_core.so (Required is at least version "3.0")
-- Found SOCIMySQL: /usr/lib64/libsoci_mysql.so (Required is at least version "3.0")
-- Found SOCI with MySQL back-end support version: 3.0.0
-- Requires StdAir-0.35
-- Found StdAir version: 0.36.2
-- Requires Doxygen without specifying any version
-- Found Doxygen: /usr/bin/doxygen
-- Found DoxygenWrapper: /usr/bin/doxygen
-- Found Doxygen version: 1.7.4
-- Had to set the linker language for 'trademgenlib' to CXX
-- Had to set the linker language for 'pytrademgenlib' to CXX
-- Test 'TrademgenTest' to be built with 'DemandGenerationTestSuite.cpp'
--
-- =====
-- -----
-- ---      Project Information      ---
-- -----
-- PROJECT_NAME ..... : trademgen
-- PACKAGE_PRETTY_NAME ..... : TraDemGen
-- PACKAGE ..... : trademgen
-- PACKAGE_NAME ..... : TRADEMGEN
-- PACKAGE_BRIEF ..... : C++ Simulated Travel Demand Generation Library
-- PACKAGE_VERSION ..... : 99.99.99
-- GENERIC_LIB_VERSION ..... : 99.99.99
-- GENERIC_LIB_SOVERSION ..... : 99.99
--
-- -----
-- ---      Build Configuration      ---
-- -----
-- Modules to build ..... : trademgen
-- Libraries to build/install ..... : trademgenlib;pytrademgenlib
-- Binaries to build/install ..... : trademgen;trademgen_with_db;pytrademgen.py
-- Modules to test ..... : trademgen
-- Binaries to test ..... : TrademgenTesttst
--
-- * Module ..... : trademgen
--   + Layers to build ..... : .;basic;bom;factory;command;service
--   + Dependencies on other layers :
--   + Libraries to build/install . : trademgenlib;pytrademgenlib
--   + Executables to build/install : trademgen;trademgen_with_db;pytrademgen.py
--   + Tests to perform ..... : TrademgenTesttst
--
-- BUILD_SHARED_LIBS ..... : ON
-- CMAKE_BUILD_TYPE ..... : Debug
-- * CMAKE_C_FLAGS ..... :
-- * CMAKE_CXX_FLAGS ..... : -Wall -Werror
-- * BUILD_FLAGS ..... :
-- * COMPILE_FLAGS ..... :
-- CMAKE_MODULE_PATH ..... : /home/user/dev/sim/trademgen/trademgengithub/config/

```

```

-- CMAKE_INSTALL_PREFIX ..... : /home/user/dev/deliveries/trademgen-99.99.99
--
-- * Doxygen:
--   - DOXYGEN_VERSION ..... : 1.7.4
--   - DOXYGEN_EXECUTABLE ..... : /usr/bin/doxygen
--   - DOXYGEN_DOT_EXECUTABLE ..... : /usr/bin/dot
--   - DOXYGEN_DOT_PATH ..... : /usr/bin
--
-- -----
-- --- Installation Configuration ---
-- -----
-- INSTALL_LIB_DIR ..... : /home/user/dev/deliveries/trademgen-99.99.99/lib64
-- INSTALL_BIN_DIR ..... : /home/user/dev/deliveries/trademgen-99.99.99/bin
-- INSTALL_INCLUDE_DIR ..... : /home/user/dev/deliveries/trademgen-99.99.99/include
-- INSTALL_DATA_DIR ..... : /home/user/dev/deliveries/trademgen-99.99.99/share
-- INSTALL_SAMPLE_DIR ..... : /home/user/dev/deliveries/trademgen-99.99.99/share/trademgen/samples
-- INSTALL_DOC ..... : ON
--
-- -----
-- --- Packaging Configuration ---
-- -----
-- CPACK_PACKAGE_CONTACT ..... : Denis Arnaud <denis_arnaud - at - users dot sourceforge dot net>
-- CPACK_PACKAGE_VENDOR ..... : Denis Arnaud
-- CPACK_PACKAGE_VERSION ..... : 99.99.99
-- CPACK_PACKAGE_DESCRIPTION_FILE . : /home/user/dev/sim/trademgen/trademgengithub/README
-- CPACK_RESOURCE_FILE_LICENSE .... : /home/user/dev/sim/trademgen/trademgengithub/COPYING
-- CPACK_GENERATOR ..... : TBZ2
-- CPACK_DEBIAN_PACKAGE_DEPENDS ... :
-- CPACK_SOURCE_GENERATOR ..... : TBZ2;TGZ
-- CPACK_SOURCE_PACKAGE_FILE_NAME . : trademgen-99.99.99
--
-- -----
-- --- External libraries ---
-- -----
--
-- * Python:
--   - PYTHONLIBS_VERSION ..... : 2.7
--   - PYTHON_LIBRARIES ..... : /usr/lib64/libpython2.7.so
--   - PYTHON_INCLUDE_PATH ..... : /usr/include/python2.7
--   - PYTHON_INCLUDE_DIRS ..... : /usr/include/python2.7
--   - PYTHON_DEBUG_LIBRARIES ..... :
--   - Python_ADDITIONAL_VERSIONS . :
--
-- * Boost:
--   - Boost_VERSION ..... : 104600
--   - Boost_LIB_VERSION ..... : 1_46
--   - Boost_HUMAN_VERSION ..... : 1.46.0
--   - Boost_INCLUDE_DIRS ..... : /usr/include
--   - Boost required components .. : program_options;date_time;iostreams;serialization;filesystem;unit_te
--   - Boost required libraries ... : optimized;/usr/lib64/libboost_iostreams-mt.so;debug;/usr/lib64/libbo
--
-- * MySQL:
--   - MYSQL_VERSION ..... : 5.5.14
--   - MYSQL_INCLUDE_DIR ..... : /usr/include/mysql
--   - MYSQL_LIBRARIES ..... : /usr/lib64/mysql/libmysqlclient.so
--
-- * SOCI:
--   - SOCI_VERSION ..... : 3.0.0
--   - SOCI_INCLUDE_DIR ..... : /usr/include/soci
--   - SOCI_MYSQL_INCLUDE_DIR ..... : /usr/include/soci
--   - SOCI_LIBRARIES ..... : /usr/lib64/libsoci_core.so
--   - SOCI_MYSQL_LIBRARIES ..... : /usr/lib64/libsoci_mysql.so
--
-- * StdAir:
--   - STDAIR_VERSION ..... : 0.36.2
--   - STDAIR_BINARY_DIRS ..... : /home/user/dev/deliveries/stdair-0.36.2/bin
--   - STDAIR_EXECUTABLES ..... : stdair

```

```
-- - STDAIR_LIBRARY_DIRS ..... : /home/user/dev/deliveries/stdair-0.36.2/lib64
-- - STDAIR_LIBRARIES ..... : stdairlib;stdairuicllib
-- - STDAIR_INCLUDE_DIRS ..... : /home/user/dev/deliveries/stdair-0.36.2/include
-- - STDAIR_SAMPLE_DIR ..... : /home/user/dev/deliveries/stdair-0.36.2/share/stdair/samples
--
-- Change a value with: cmake -D<Variable>=<Value>
-- =====
--
-- Configuring done
-- Generating done
-- Build files have been written to: /home/user/dev/sim/trademgen/trademgengithub/build
```

It is recommended that you check if your library has been compiled and linked properly and works as expected. To do so, you should execute the testing process 'make check'. As a result, you should obtain a similar report:

```
[ 0%] Built target hdr_cfg_trademgen
[ 94%] Built target trademgenlib
[100%] Built target TrademgenTesttst
Scanning dependencies of target check_trademgentst
Test project /home/user/dev/sim/trademgen/trademgengithub/build/test/trademgen
  Start 1: TrademgenTesttst
1/1 Test #1: TrademgenTesttst ..... Passed    0.37 sec

100% tests passed, 0 tests failed out of 1

Total Test time (real) = 10.82 sec
[100%] Built target check_trademgentst
Scanning dependencies of target check
[100%] Built target check
```

Check if all the executed tests PASSED. If not, please contact us by filling a [bug-report](#).

Finally, you should install the compiled and linked library, include files and (optionally) HTML and PDF documentation by typing:

```
make install
```

Depending on the PREFIX settings during configuration, you might need the root (administrator) access to perform this step.

Eventually, you might invoke the following command

```
make clean
```

to remove all files created during compilation process, or even

```
cd ~/dev/sim/trademgengit
rm -rf build && mkdir build
cd build
```

to remove everything.

## 12.9 Linking with TraDemGen

### 12.9.1 Table of Contents

- [Introduction](#)
- [Using the pkg-config command](#)
- [Using the trademgen-config script](#)
- [M4 macro for the GNU Autotools](#)
- [Using TraDemGen with dynamic linking](#)

### 12.9.2 Introduction

There are two convenient methods of linking your programs with the TraDemGen library. The first one employs the 'pkg-config' command (see <http://pkgconfig.freedesktop.org/>), whereas the second one uses 'trademgen-config' script. These methods are shortly described below.

### 12.9.3 Using the pkg-config command

'pkg-config' is a helper tool used when compiling applications and libraries. It helps you insert the correct compiler and linker options. The syntax of the 'pkg-config' is as follows:

```
pkg-config <options> <library_name>
```

For instance, assuming that you need to compile an TraDemGen based program 'my\_prog.cpp', you should use the following command:

```
g++ `pkg-config --cflags trademgen` -o my_prog my_prog.cpp `pkg-config --libs trademgen`
```

For more information see the 'pkg-config' man pages.

### 12.9.4 Using the trademgen-config script

TraDemGen provides a shell script called 'trademgen-config', which is installed by default in '\$prefix/bin' ('/usr/local/bin') directory. It can be used to simplify compilation and linking of TraDemGen based programs. The usage of this script is quite similar to the usage of the 'pkg-config' command.

Assuming that you need to compile the program 'my\_prog.cpp' you can now do that with the following command:

```
g++ `trademgen-config --cflags` -o my_prog_opt my_prog.cpp `trademgen-config --libs`
```

A list of 'trademgen-config' options can be obtained by typing:

```
trademgen-config --help
```

If the 'trademgen-config' command is not found by your shell, you should add its location '\$prefix/bin' to the PATH environment variable, e.g.:

```
export PATH=/usr/local/bin:$PATH
```



### 12.9.5 M4 macro for the GNU Autotools

A M4 macro file is delivered with TraDemGen, namely 'trademgen.m4', which can be found in, e.g., '/usr/share/aclocal'. When used by a 'configure' script, thanks to the 'AM\_PATH\_TraDemGen' macro (specified in the M4 macro file), the following Makefile variables are then defined:

- 'TraDemGen\_VERSION' (e.g., defined to 0.23.0)
- 'TraDemGen\_CFLAGS' (e.g., defined to '-I\${prefix}/include')
- 'TraDemGen\_LIBS' (e.g., defined to '-L\${prefix}/lib -ltrademgen')

### 12.9.6 Using TraDemGen with dynamic linking

When using static linking some of the library routines in TraDemGen are copied into your executable program. This can lead to unnecessary large executables. To avoid having too large executable files you may use dynamic linking instead. Dynamic linking means that the actual linking is performed when the program is executed. This requires that the system is able to locate the shared TraDemGen library file during your program execution. If you install the TraDemGen library using a non-standard prefix, the 'LD\_LIBRARY\_PATH' environment variable might be used to inform the linker of the dynamic library location, e.g.:

```
export LD_LIBRARY_PATH=<TraDemGen installation prefix>/lib:$LD_LIBRARY_PATH
```

## 12.10 Test Rules

This section describes how the functionality of the TraDemGen library should be verified. In the 'test/trademgen' subdirectory, test source files are provided. All functionality should be tested using these test source files.

### 12.10.1 The Test Source Files

Each new TraDemGen module/class should be accompanied with a test source file. The test source file is an implementation in C++ that tests the functionality of a function/class or a group of functions/classes called test suites. The test source file should test relevant parameter settings and input/output relations to guarantee correct functionality of the corresponding classes/functions. The test source files should be maintained using version control and updated whenever new functionality is added to the TraDemGen library.

The test source file should print relevant data to a standard output that can be used to verify the functionality. All relevant parameter settings should be tested.

The test source file should be placed in the 'test/trademgen' subdirectory and should have a name ending with 'TestSuite.cpp'.

### 12.10.2 The Reference File

Consider a test source file named 'YieldTestSuite.cpp'. A reference file named 'YieldTestSuite.ref' should accompany the test source file. The reference file contains a reference printout of the standard output generated when running the test program. The reference file should be maintained using version control and updated according to the test source file.

### 12.10.3 Testing TraDemGen Library

One can compile and execute all test programs from the ``test/trademgen'` sub-directory by typing:

```
% make check
```

after successful compilation of the TraDemGen library.

## 12.11 Users Guide

### 12.11.1 Table of Contents

- [Introduction](#)
- [Get Started](#)
  - [Get the TraDemGen library](#)
  - [Build the TraDemGen project](#)
  - [Build and Run the Tests](#)
  - [Install the TraDemGen Project \(Binaries, Documentation\)](#)
- [Exploring the Predefined BOM Tree](#)
  - [Demand Stream Engine BOM Tree](#)
- [Extending the BOM Tree](#)

### 12.11.2 Introduction

The TraDemGen library contains classes for yield rule management. This document does not cover all the aspects of the TraDemGen library. It does however explain the most important things you need to know in order to start using TraDemGen.

### 12.11.3 Get Started

#### 12.11.3.1 Get the TraDemGen library

**12.11.3.2 Build the TraDemGen project** To run the configuration script the first time, go to the top directory (where the TraDemGen package has been un-packed), and issue either of the following two commands, depending on whether the TraDemGen project has been checked out from the Subversion repository or downloaded as a tar-ball package from the Sourceforge Web site:

- `./autogen.sh`
- `./configure`

#### 12.11.3.3 Build and Run the Tests

#### 12.11.3.4 Install the TraDemGen Project (Binaries, Documentation)

### 12.11.4 Exploring the Predefined BOM Tree

TraDemGen predefines a BOM (Business Object Model) tree specific to the airline IT arena.

#### 12.11.4.1 Demand Stream Engine BOM Tree

- [TRADEMGEN::DemandStream](#)

### 12.11.5 Extending the BOM Tree

## 12.12 Supported Systems

### 12.12.1 Table of Contents

- [Introduction](#)
- [TraDemGen 3.10.x](#)
  - [Linux Systems](#)
    - \* [Fedora Core 4 with ATLAS](#)
    - \* [Gentoo Linux with ACML](#)
    - \* [Gentoo Linux with ATLAS](#)
    - \* [Gentoo Linux with MKL](#)
    - \* [Gentoo Linux with NetLib's BLAS and LAPACK](#)
    - \* [Red Hat Enterprise Linux with TraDemGen External](#)
    - \* [SUSE Linux 10.0 with NetLib's BLAS and LAPACK](#)
    - \* [SUSE Linux 10.0 with MKL](#)
  - [Windows Systems](#)
    - \* [Microsoft Windows XP with Cygwin](#)
    - \* [Microsoft Windows XP with Cygwin and ATLAS](#)
    - \* [Microsoft Windows XP with Cygwin and ACML](#)
    - \* [Microsoft Windows XP with MinGW, MSYS and ACML](#)
    - \* [Microsoft Windows XP with MinGW, MSYS and TraDemGen External](#)
    - \* [Microsoft Windows XP with MS Visual C++ and Intel MKL](#)
  - [Unix Systems](#)
    - \* [SunOS 5.9 with TraDemGen External](#)
- [TraDemGen 3.9.1](#)
- [TraDemGen 3.9.0](#)
- [TraDemGen 3.8.1](#)

### 12.12.2 Introduction

This page is intended to provide a list of TraDemGen supported systems, i.e. the systems on which configuration, installation and testing process of the TraDemGen library has been successful. Results are grouped based on minor release number. Therefore, only the latest tests for bug-fix releases are included. Besides, the information on this page is divided into sections dependent on the operating system.

Where necessary, some extra information is given for each tested configuration, e.g. external libraries installed, configuration commands used, etc.

If you manage to compile, install and test the TraDemGen library on a system not mentioned below, please let us know, so we could update this database.

### 12.12.3 TraDemGen 3.10.x

#### 12.12.3.1 Linux Systems

##### Fedora Core 4 with ATLAS

- **Platform:** Intel Pentium 4
- **Operating System:** Fedora Core 4 (x86)
- **Compiler:** g++ (GCC) 4.0.2 20051125
- **TraDemGen release:** 3.10.0
- **External Libraries:** From FC4 distribution:
  - fftw3.i386-3.0.1-3
  - fftw3-devel.i386-3.0.1-3
  - atlas-sse2.i386-3.6.0-8.fc4
  - atlas-sse2-devel.i386-3.6.0-8.fc4
  - blas.i386-3.0-35.fc4
  - lapack.i386-3.0-35.fc4
- **Tests Status:** All tests PASSED
- **Comments:** TraDemGen configured with:

```
% CXXFLAGS="-O3 -pipe -march=pentium4" ./configure
```
- **Date:** March 7, 2006
- **Tester:** Tony Ottosson

##### Gentoo Linux with ACML

- **Platform:** AMD Sempron 3000+
- **Operating System:** Gentoo Linux 2006.0 (x86 arch)
- **Compiler(s):** g++ (GCC) 3.4.5
- **TraDemGen release:** 3.10.1
- **External Libraries:** Compiled and installed from portage tree:
  - sci-libs/acml-3.0.0
- **Tests Status:** All tests PASSED
- **Comments:** BLAS and LAPACK libs set by using the following system commands:

```
% eselect blas set ACML
% eselect lapack set ACML
```

TraDemGen configured with:

```
% export CPPFLAGS="-I/usr/include/acml"
% ./configure --with-blas="-lblas"
```

- **Date:** March 31, 2006
- **Tester:** Adam Piatyszek (ediap)

### Gentoo Linux with ATLAS

- **Platform:** Intel Pentium M Centrino
- **Operating System:** Gentoo Linux 2006.0 (x86)
- **Compiler:** g++ (GCC) 3.4.5
- **TraDemGen release:** 3.10.1
- **External Libraries:** Compiled and installed from portage tree:
  - sci-libs/fftw-3.1
  - sci-libs/blas-atlas-3.6.0-r1
  - sci-libs/lapack-atlas-3.6.0
- **Tests Status:** All tests PASSED
- **Comments:** BLAS and LAPACK libs set by using the following system commands:

```
% eselect blas set ATLAS
% eselect lapack set ATLAS
```

TraDemGen configured with:

```
% ./configure --with-blas="-lblas"
```

- **Date:** March 31, 2006
- **Tester:** Adam Piatyszek (ediap)

### Gentoo Linux with MKL

- **Platform:** Intel Pentium M Centrino
- **Operating System:** Gentoo Linux 2006.0 (x86 arch)
- **Compiler:** g++ (GCC) 3.4.5
- **TraDemGen release:** 3.10.0
- **External Libraries:** Intel Math Kernel Library (MKL) 8.0.1 installed manually in the following directory: /opt/intel/mkl/8.0.1
- **Tests Status:** All tests PASSED

- **Comments:** TraDemGen configured using the following commands:

```
% export LDFLAGS="-L/opt/intel/mkl/8.0.1/lib/32"
% export CPPFLAGS="-I/opt/intel/mkl/8.0.1/include"
% ./configure
```

- **Date:** February 28, 2006
- **Tester:** Adam Piatyszek (ediap)

### Gentoo Linux with NetLib's BLAS and LAPACK

- **Platform:** Intel Pentium M Centrino
- **Operating System:** Gentoo Linux 2006.0 (x86)
- **Compiler:** g++ (GCC) 3.4.5
- **TraDemGen release:** 3.10.1
- **External Libraries:** Compiled and installed from portage tree:
  - sci-libs/fftw-3.1
  - sci-libs/blas-reference-19940131-r2
  - sci-libs/cblas-reference-20030223
  - sci-libs/lapack-reference-3.0-r2
- **Tests Status:** All tests PASSED
- **Comments:** BLAS and LAPACK libs set by using the following system commands:

```
% blas-config reference
% lapack-config reference
```

TraDemGen configured with:

```
% ./configure --with-blas="-lblas"
```

- **Date:** March 31, 2006
- **Tester:** Adam Piatyszek (ediap)

### Red Hat Enterprise Linux with TraDemGen External

- **Platform:** Intel Pentium 4
- **Operating System:** Red Hat Enterprise Linux AS release 4 (Nahant Update 2)
- **Compiler:** g++ (GCC) 3.4.4 20050721 (Red Hat 3.4.4-2)
- **TraDemGen release:** 3.10.0
- **External Libraries:** BLAS, CBLAS, LAPACK and FFTW libraries from TraDemGen External 2.1.1 package
- **Tests Status:** All tests PASSED
- **Date:** March 7, 2006
- **Tester:** Erik G. Larsson

**SUSE Linux 10.0 with NetLib's BLAS and LAPACK**

- **Platform:** Intel Pentium 4 CPU 3.20GHz (64-bit)
- **Operating System:** SUSE Linux 10.0 (x86\_64)
- **Compiler(s):** g++ (GCC) 4.0.2
- **TraDemGen release:** 3.10.0
- **External Libraries:** BLAS, LAPACK and FFTW libraries installed from OpenSuse 10.0 RPM repository:
  - blas-3.0-926
  - lapack-3.0-926
  - fftw3-3.0.1-114
  - fftw3-threads-3.0.1-114
  - fftw3-devel-3.0.1-114
- **Tests Status:** All tests PASSED
- **Comments:** TraDemGen configured with:

```
% export CXXFLAGS="-m64 -march=nocona -O3 -pipe"
% ./configure --with-lapack="/usr/lib64/liblapack.so.3"
```
- **Date:** March 1, 2006
- **Tester:** Adam Piatyszek (ediap)

**SUSE Linux 10.0 with MKL**

- **Platform:** Intel Pentium 4 CPU 3.20GHz (64-bit)
- **Operating System:** SUSE Linux 10.0 (x86\_64)
- **Compiler(s):** g++ (GCC) 4.0.2
- **TraDemGen release:** 3.10.0
- **External Libraries:** Intel Math Kernel Library (MKL) 8.0.1 installed manually in the following directory: /opt/intel/mkl/8.0.1
- **Tests Status:** All tests PASSED
- **Comments:** TraDemGen configured with:

```
% export CXXFLAGS="-m64 -march=nocona -O3 -pipe"
% export LDFLAGS="-L/opt/intel/mkl/8.0.1/lib/em64t"
% export CPPFLAGS="-I/opt/intel/mkl/8.0.1/include"
% ./configure
```
- **Date:** March 1, 2006
- **Tester:** Adam Piatyszek (ediap)

**12.12.3.2 Windows Systems**

**Microsoft Windows XP with Cygwin**

- **Platform:** AMD Sempron 3000+
- **Operating System:** Microsoft Windows XP SP2, Cygwin 1.5.19-4
- **Compiler(s):** g++ (GCC) 3.4.4 (cygming special)
- **TraDemGen release:** 3.10.1
- **External Libraries:** Installed from Cygwin's repository:
  - `fftw-3.0.1-2`
  - `fftw-dev-3.0.1-1`
  - `lapack-3.0-4`
- **Tests Status:** All tests PASSED
- **Comments:** Only static library can be built. TraDemGen configured with:  

```
% ./configure
```
- **Date:** March 31, 2006
- **Tester:** Adam Piatyszek (ediap)

**Microsoft Windows XP with Cygwin and ATLAS**

- **Platform:** AMD Sempron 3000+
- **Operating System:** Microsoft Windows XP SP2, Cygwin 1.5.19-4
- **Compiler(s):** g++ (GCC) 3.4.4 (cygming special)
- **TraDemGen release:** 3.10.1
- **External Libraries:** Installed from Cygwin's repository:
  - `fftw-3.0.1-2`
  - `fftw-dev-3.0.1-1`ATLAS BLAS and LAPACK libraries from TraDemGen External 2.1.1 package configured using:  

```
% ./configure --enable-atlas --disable-fftw
```
- **Tests Status:** All tests PASSED
- **Comments:** Only static library can be built. TraDemGen configured with:  

```
% export LDFLAGS="-L/usr/local/lib"
% ./configure
```
- **Date:** March 31, 2006
- **Tester:** Adam Piatyszek (ediap)



**Microsoft Windows XP with Cygwin and ACML**

- **Platform:** AMD Sempron 3000+
- **Operating System:** Microsoft Windows XP SP2, Cygwin 1.5.19-4
- **Compiler(s):** g++ (GCC) 3.4.4 (cygming special)
- **TraDemGen release:** 3.10.2
- **External Libraries:** ACML version 3.1.0 (acml3.1.0-32-win32-g77.exe) installed into a default directory, i.e. "c:\Program Files\AMD\acml3.1.0"
- **Tests Status:** All tests PASSED
- **Comments:** Only static library can be built. TraDemGen configured with:

```
% export LDFLAGS="-L/cygdrive/c/Progra~1/AMD/acml3.1.0/gnu32/lib"
% export CPPFLAGS="-I/cygdrive/c/Progra~1/AMD/acml3.1.0/gnu32/include"
% ./configure --enable-debug
```

- **Date:** May 15, 2006
- **Tester:** Adam Piatyszek (ediap)

**Microsoft Windows XP with MinGW, MSYS and ACML**

- **Platform:** AMD Sempron 3000+
- **Operating System:** Microsoft Windows XP SP2, MinGW 5.0.2, MSYS 1.0.10
- **Compiler(s):** g++ (GCC) 3.4.4 (mingw special)
- **TraDemGen release:** 3.10.2
- **External Libraries:** ACML version 3.1.0 (acml3.1.0-32-win32-g77.exe) installed into a default directory, i.e. "c:\Program Files\AMD\acml3.1.0"
- **Tests Status:** All tests PASSED
- **Comments:** Only static library can be built. TraDemGen configured with:

```
% export LDFLAGS="-L/c/Progra~1/AMD/acml3.1.0/gnu32/lib"
% export CPPFLAGS="-I/c/Progra~1/AMD/acml3.1.0/gnu32/include"
% ./configure --enable-debug
```

- **Date:** May 15, 2006
- **Tester:** Adam Piatyszek (ediap)

**Microsoft Windows XP with MinGW, MSYS and TraDemGen External**

- **Platform:** AMD Sempron 3000+
- **Operating System:** Microsoft Windows XP SP2, MinGW 5.0.2, MSYS 1.0.10
- **Compiler(s):** g++ (GCC) 3.4.4 (mingw special)
- **TraDemGen release:** 3.10.5
- **External Libraries:** BLAS, CBLAS, LAPACK and FFTW libraries from TraDemGen External 2.2.0 package
- **Tests Status:** All tests PASSED
- **Comments:** Only static library can be built. TraDemGen configured with:

```
% export LDFLAGS="-L/usr/local/lib"
% export CPPFLAGS="-I/usr/local/include"
% export CXXFLAGS="-Wall -O3 -march=athlon-tbird -pipe"
% ./configure --disable-html-doc
```

- **Date:** August 11, 2006
- **Tester:** Adam Piatyszek (ediap)

**Microsoft Windows XP with MS Visual C++ and Intel MKL**

- **Platform:** AMD Sempron 3000+
- **Operating System:** Microsoft Windows XP SP2
- **Compiler(s):** Microsoft Visual C++ 2005 .NET
- **TraDemGen release:** 3.10.5
- **External Libraries:** Intel Math Kernel Library (MKL) 8.1 installed manually in the following directory: "C:\Program Files\Intel\MKL\8.1"
- **Tests Status:** Not fully tested. Some TraDemGen based programs compiled and run with success.
- **Comments:** Only static library can be built. TraDemGen built by opening the "win32\trademgen.vcproj" project file in MSVC++ and executing "Build → Build Solution" command from menu.
- **Date:** August 11, 2006
- **Tester:** Adam Piatyszek (ediap)

**12.12.3.3 Unix Systems**

**SunOS 5.9 with TraDemGen External**

- **Platform:** SUNW, Sun-Blade-100 (SPARC)
- **Operating System:** SunOS 5.9 Generic\_112233-10
- **Compiler(s):** g++ (GCC) 3.4.5
- **TraDemGen release:** 3.10.2
- **External Libraries:** BLAS, CBLAS, LAPACK and FFTW libraries from TraDemGen External 2.1.1 package. The following configuration command has been used:

```
% export CFLAGS="-mcpu=ultrasparc -O2 -pipe -funroll-all-loops"  
% ./configure
```

- **Tests Status:** All tests PASSED
- **Comments:** TraDemGen configured with:

```
% export LDFLAGS="-L/usr/local/lib"  
% export CPPFLAGS="-I/usr/local/include"  
% export CXXFLAGS="-mcpu=ultrasparc -O2 -pipe"  
% ./configure --enable-debug
```

- **Date:** May 15, 2006
- **Tester:** Adam Piatyszek (ediap)

**12.13 TraDemGen Supported Systems (Previous Releases)****12.13.1 TraDemGen 3.9.1****12.13.2 TraDemGen 3.9.0****12.13.3 TraDemGen 3.8.1****12.14 Tutorials****12.14.1 Table of Contents**

- [Introduction](#)
  - [Preparing the StdAir Project for Development](#)
- [Build a Predefined BOM Tree](#)
  - [Instantiate the BOM Root Object](#)
  - [Instantiate the \(Airline\) Inventory Object](#)
  - [Link the Inventory Object with the BOM Root](#)
  - [Build Another Airline Inventory](#)
  - [Dump The BOM Tree Content](#)
  - [Result of the Tutorial Program](#)
- [Extend the Pre-Defined BOM Tree](#)
  - [Extend an Airline Inventory Object](#)
  - [Build the Specific BOM Objects](#)
  - [Result of the Tutorial Program](#)

### 12.14.2 Introduction

This page contains some tutorial examples that will help you getting started using StdAir. Most examples show how to construct some simple business objects, i.e., instances of the so-named Business Object Model (BOM).

**12.14.2.1 Preparing the StdAir Project for Development** The source code for these examples can be found in the `batches` and `test/stdair` directories. They are compiled along with the rest of the StdAir project. See the User Guide ([Users Guide](#)) for more details on how to build the StdAir project.

### 12.14.3 Build a Predefined BOM Tree

A few steps:

- [Instantiate the BOM Root Object](#)
- [Instantiate the \(Airline\) Inventory Object](#)
- [Link the Inventory Object with the BOM Root](#)

**12.14.3.1 Instantiate the BOM Root Object** First, a BOM root object (i.e., a root for all the classes in the project) is instantiated by the `stdair::STDAIR_ServiceContext` context object, when the `stdair::STDAIR_Service` is itself instantiated. The corresponding StdAir type (class) is `stdair::BomRoot`.

In the following sample, that object is named `ioBomRoot`, and is given as input/output parameter of the `stdair::CmdBomManager::buildSampleBom()` method:

**12.14.3.2 Instantiate the (Airline) Inventory Object** An airline inventory object can then be instantiated. Let us give it the "BA" airline code (corresponding to British Airways) as the object key. That is, an object (let us name it `lBAKey`) of type (class) `stdair::InventoryKey` has first to be instantiated.

Thanks to that key, an airline inventory object, i.e. of type (class) `stdair::Inventory`, can be instantiated. Let us name that airline inventory object `lBAInv`.

**12.14.3.3 Link the Inventory Object with the BOM Root** Then, both objects have to be linked: the airline inventory object (`stdair::Inventory`) has to be linked with the root of the BOM tree (`stdair::BomRoot`). That operation is as simple as using the `stdair::FacBomManager::addToListAndMap()` method:

**12.14.3.4 Build Another Airline Inventory** Another airline inventory object, corresponding to the Air France (Air France) company, is instantiated the same way:

See the corresponding full program (`cmd_bom_manager_cpp`) for more details.

**12.14.3.5 Dump The BOM Tree Content** From the `BomRoot` (of type `stdair::BomRoot`) object instance, the list of airline inventories (of type `stdair::Inventory`) can then be retrieved...

... and browsed:

See the corresponding full program (`bom_display_cpp`) for more details.

**12.14.3.6 Result of the Tutorial Program** When the `stdair.cpp` program is run (with the `-b` option), the output should look like:

See the corresponding full program (`batch_stdair_cpp`) for more details.

## 12.14.4 Extend the Pre-Defined BOM Tree

Now that we master how to instantiate the pre-defined `StdAir` classes, let us see how to extend that BOM.

**12.14.4.1 Extend an Airline Inventory Object** For instance, let us assume that some (IT) provider (e.g., you) would like to have a specific implementation of the `Inventory` object. The corresponding class has just to extend the `stdair::Inventory` class:

The STL containers have to be defined accordingly too:

See the full class definition (`test_archi_inv_hpp`) and implementation (`test_archi_inv_cpp`) for more details.

**12.14.4.2 Build the Specific BOM Objects** The BOM root object (`stdair::BomRoot`) is instantiated the classical way:

Then, the specific implementation of the airline inventory object (`myprovider::Inventory`) can be instantiated the same way as a standard `Inventory` (`stdair::Inventory`) would be:

Then, the specific implementation of the airline inventory object (`myprovider::Inventory`) is linked to the root of the BOM tree (`stdair::BomRoot`) the same way as the standard Inventory (`stdair::Inventory`) would be:

Another specific airline inventory object is instantiated the same way:

From the `BomRoot` (of type `stdair::BomRoot`) object instance, the list of specific airline inventories (of type `stdair::Inventory`) can then be retrieved...

... and browsed:

**12.14.4.3 Result of the Tutorial Program** When this program is run, the output should look like:

See the corresponding full program (`StandardAirlineITTestSuite.cpp`) for more details.

## 12.15 Command-Line Test to Demonstrate How To Use TraDemGen elements

```

*/
// //////////////////////////////////////
// Import section
// //////////////////////////////////////
// STL
#include <sstream>
#include <fstream>
#include <map>
#include <cmath>
// Boost Unit Test Framework (UTF)
#define BOOST_TEST_DYN_LINK
#define BOOST_TEST_MAIN
#define BOOST_TEST_MODULE DemandGenerationTest
#include <boost/test/unit_test.hpp>
// StdAir
#include <stdair/stdair_basic_types.hpp>
#include <stdair/basic/BasConst_General.hpp>
#include <stdair/basic/BasLogParams.hpp>
#include <stdair/basic/BasDBParams.hpp>
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/basic/ProgressStatusSet.hpp>
#include <stdair/bom/EventStruct.hpp>
#include <stdair/bom/BookingRequestStruct.hpp>
#include <stdair/service/Logger.hpp>
// TraDemGen
#include <trademgen/TRADEMGEN_Exceptions.hpp>
#include <trademgen/TRADEMGEN_Service.hpp>

```

```

#include <trademgen/bom/DemandStreamKey.hpp>
#include <trademgen/config/trademgen-paths.hpp>

namespace boost_utf = boost::unit_test;

// (Boost) Unit Test XML Report
std::ofstream utfReportStream ("DemandGenerationTestSuite_utfresults.xml");

struct UnitTestConfig {
    UnitTestConfig() {
        boost_utf::unit_test_log.set_stream (utfReportStream);
        boost_utf::unit_test_log.set_format (boost_utf::XML);
        boost_utf::unit_test_log.set_threshold_level (boost_utf::log_test_units);
        //boost_utf::unit_test_log.set_threshold_level (boost_utf::log_successful_tests);
    }

    ~UnitTestConfig() {
    }
};

// Specific type definitions
typedef std::pair<stdair::Count_T, stdair::Count_T> NbOfEventsPair_T;
typedef std::map<const stdair::DemandStreamKeyStr_T,
                NbOfEventsPair_T> NbOfEventsByDemandStreamMap_T;

// //////////////////////////////////////
void testDemandGenerationHelper (const unsigned short iTestFlag,
                                const stdair::Filename_T& iDemandInputFilename,
                                const stdair::DemandGenerationMethod& iDemandGenerationMethod,
                                const bool isBuiltin) {

    // Seed for the random generation
    const stdair::RandomSeed_T lRandomSeed = stdair::DEFAULT_RANDOM_SEED;

    // Output log File
    std::ostringstream oStr;
    oStr << "DemandGenerationTestSuite_" << iTestFlag << ".log";
    const stdair::Filename_T lLogFilename (oStr.str());

    // Set the log parameters
    std::ofstream logOutputFile;
    // Open and clean the log outputfile
    logOutputFile.open (lLogFilename.c_str());
    logOutputFile.clear();

    // Initialise the TraDemGen service object
    const stdair::BasLogParams lLogParams (stdair::LOG::DEBUG, logOutputFile);
    TRADEMGEN::TRADEMGEN_Service trademgenService (lLogParams, lRandomSeed);

    NbOfEventsByDemandStreamMap_T lNbOfEventsMap;

    // Total number of events
    stdair::Count_T lRefExpectedNbOfEvents (0);
    stdair::Count_T lRefActualNbOfEvents (0);

    // Check whether or not a (CSV) input file should be read
    if (isBuiltin == true) {

        // Build the default sample BOM tree (filled with demand streams) for TraDemGen
        trademgenService.buildSampleBom();

        lNbOfEventsMap.insert (NbOfEventsByDemandStreamMap_T::
                                value_type ("SIN-BKK 2010-Feb-08 Y",
                                              NbOfEventsPair_T (4, 60)));
        lNbOfEventsMap.insert (NbOfEventsByDemandStreamMap_T::
                                value_type ("BKK-HKG 2010-Feb-08 Y",
                                              NbOfEventsPair_T (4, 60)));
    }
}

```

```

lNbOfEventsMap.insert (NbOfEventsByDemandStreamMap_T::
                        value_type ("SIN-HKG 2010-Feb-08 Y",
                                    NbOfEventsPair_T (4, 60)));

// Total number of events, for the 3 demand streams: 180
lRefExpectedNbOfEvents = 180;
lRefActualNbOfEvents = 186;

} else {

// Create the DemandStream objects, and insert them within the BOM tree
const TRADEMGEN::DemandFilePath lDemandFilePath (iDemandInputFilename);
trademngenService.parseAndLoad (lDemandFilePath);

lNbOfEventsMap.insert (NbOfEventsByDemandStreamMap_T::
                        value_type ("SIN-HND 2010-Feb-08 Y",
                                    NbOfEventsPair_T (1, 10)));
lNbOfEventsMap.insert (NbOfEventsByDemandStreamMap_T::
                        value_type ("SIN-HND 2010-Feb-09 Y",
                                    NbOfEventsPair_T (1, 10)));
lNbOfEventsMap.insert (NbOfEventsByDemandStreamMap_T::
                        value_type ("SIN-BKK 2010-Feb-08 Y",
                                    NbOfEventsPair_T (1, 10)));
lNbOfEventsMap.insert (NbOfEventsByDemandStreamMap_T::
                        value_type ("SIN-BKK 2010-Feb-09 Y",
                                    NbOfEventsPair_T (1, 10)));

// Total number of events, for the 4 demand streams: 40
lRefExpectedNbOfEvents = 40;
lRefActualNbOfEvents = 40;
}

// Retrieve the expected (mean value of the) number of events to be
// generated
const stdair::Count_T& lExpectedNbOfEventsToBeGenerated =
    trademngenService.getExpectedTotalNumberOfRequestsToBeGenerated();

BOOST_CHECK_EQUAL (lRefExpectedNbOfEvents,
                    std::floor (lExpectedNbOfEventsToBeGenerated));

BOOST_CHECK_MESSAGE (lRefExpectedNbOfEvents ==
                    std::floor (lExpectedNbOfEventsToBeGenerated),
                    "Expected total number of requests to be generated: "
                    << lExpectedNbOfEventsToBeGenerated
                    << " (=) "
                    << std::floor (lExpectedNbOfEventsToBeGenerated)
                    << "). Reference value: " << lRefExpectedNbOfEvents);

const stdair::Count_T& lActualNbOfEventsToBeGenerated =
    trademngenService.generateFirstRequests(iDemandGenerationMethod);

// DEBUG
STDAIR_LOG_DEBUG ("Expected number of events: "
                  << lExpectedNbOfEventsToBeGenerated << ", actual: "
                  << lActualNbOfEventsToBeGenerated);

// Total number of events, for all the demand streams:
BOOST_CHECK_EQUAL (lRefActualNbOfEvents, lActualNbOfEventsToBeGenerated);

BOOST_CHECK_MESSAGE (lRefActualNbOfEvents == lActualNbOfEventsToBeGenerated,
                    "Actual total number of requests to be generated: "
                    << lExpectedNbOfEventsToBeGenerated
                    << " (=) "
                    << std::floor (lExpectedNbOfEventsToBeGenerated)
                    << "). Reference value: " << lRefActualNbOfEvents);

const bool isQueueDone = trademngenService.isQueueDone();

```



```

BOOST_REQUIRE_MESSAGE (isQueueDone == false,
    "The event queue should not be empty.");

stdair::Count_T idx = 1;
while (trademgenService.isQueueDone() == false) {

    // Get the next event from the event queue
    stdair::EventStruct lEventStruct;
    stdair::ProgressStatusSet lPPS = trademgenService.popEvent (lEventStruct);

    // DEBUG
    STDAIR_LOG_DEBUG ("Poped event: '" << lEventStruct.describe() << "'.");

    // Extract the corresponding demand/booking request
    const stdair::BookingRequestStruct& lPoppedRequest =
        lEventStruct.getBookingRequest();

    // DEBUG
    STDAIR_LOG_DEBUG ("Poped booking request: '"
        << lPoppedRequest.describe() << "'.");

    // Retrieve the corresponding demand stream
    const stdair::DemandGeneratorKey_T& lDemandStreamKey =
        lPoppedRequest.getDemandGeneratorKey();

    // Check that the number of booking requests to be generated are correct
    const NbOfEventsByDemandStreamMap_T::iterator itNbOfEventsMap =
        lNbOfEventsMap.find (lDemandStreamKey);
    BOOST_REQUIRE_MESSAGE (itNbOfEventsMap != lNbOfEventsMap.end(),
        "The demand stream key '" << lDemandStreamKey
        << "' is not expected in that test");

    const NbOfEventsPair_T& lNbOfEventsPair = itNbOfEventsMap->second;
    stdair::Count_T lCurrentNbOfEvents = lNbOfEventsPair.first;
    const stdair::Count_T& lExpectedTotalNbOfEvents = lNbOfEventsPair.second;

    // Assess whether more events should be generated for that demand stream
    const bool stillHavingRequestsToBeGenerated = trademgenService.
        stillHavingRequestsToBeGenerated (lDemandStreamKey, lPPS,
            iDemandGenerationMethod);

    if (lCurrentNbOfEvents == 1) {
        const stdair::ProgressStatus& lDemandStreamProgressStatus =
            lPPS.getSpecificGeneratorStatus();
        const stdair::Count_T& lNbOfRequests =
            lDemandStreamProgressStatus.getExpectedNb();

        BOOST_CHECK_EQUAL (lNbOfRequests, lExpectedTotalNbOfEvents);
        BOOST_CHECK_MESSAGE (lNbOfRequests == lExpectedTotalNbOfEvents,
            "[" << lDemandStreamKey
            << "] Total number of requests to be generated: "
            << lNbOfRequests << "). Expected value: "
            << lExpectedTotalNbOfEvents);
    }

    // DEBUG
    STDAIR_LOG_DEBUG ("=> [" << lDemandStreamKey << "]"[" << lCurrentNbOfEvents
        << "/" << lExpectedTotalNbOfEvents
        << "] is now processed. "
        << "Still generate events for that demand stream? "
        << stillHavingRequestsToBeGenerated);

    // If there are still events to be generated for that demand stream,
    // generate and add them to the event queue
    if (stillHavingRequestsToBeGenerated == true) {
        const stdair::BookingRequestPtr_T lNextRequest_ptr =
            trademgenService.generateNextRequest (lDemandStreamKey,

```

```

                                iDemandGenerationMethod);
assert (lNextRequest_ptr != NULL);

const stdair::Duration_T lDuration =
    lNextRequest_ptr->getRequestDateTime()
    - lPoppedRequest.getRequestDateTime();
BOOST_REQUIRE_GT (lDuration.total_milliseconds(), 0);
BOOST_REQUIRE_MESSAGE (lDuration.total_milliseconds() > 0,
    "[" << lDemandStreamKey
    << "]" The date-time of the generated event ("
    << lNextRequest_ptr->getRequestDateTime()
    << ") is lower than the date-time "
    << "of the current event ("
    << lPoppedRequest.getRequestDateTime() << ")");

// DEBUG
STDAIR_LOG_DEBUG ("[" << lDemandStreamKey << "]"[" << lCurrentNbOfEvents
    << "/" << lExpectedTotalNbOfEvents
    << "]" Added request: '" << lNextRequest_ptr->describe()
    << "'. Is queue done? "
    << trademgenService.isQueueDone());

// Keep, within the dedicated map, the current counters of events updated.
++lCurrentNbOfEvents;
itNbOfEventsMap->second = NbOfEventsPair_T (lCurrentNbOfEvents,
                                            lExpectedTotalNbOfEvents);
}

// Iterate
++idx;
}
// Compensate for the last iteration
--idx;

if (iDemandGenerationMethod == stdair::DemandGenerationMethod::STA_ORD) {
    //
    BOOST_CHECK_EQUAL (idx, lRefActualNbOfEvents);
    BOOST_CHECK_MESSAGE (idx == lRefActualNbOfEvents,
        "The total actual number of events is "
        << lRefActualNbOfEvents << ", but " << idx
        << " events have been generated");
}

trademgenService.reset();

// DEBUG
STDAIR_LOG_DEBUG ("End of the simulation");

// Close the log file
logOutputFile.close();
}

// ////////////////////////////////// Main: Unit Test Suite //////////////////////////////////

// Set the UTF configuration (re-direct the output to a specific file)
BOOST_GLOBAL_FIXTURE (UnitTestFixture);

// Start the test suite
BOOST_AUTO_TEST_SUITE (master_test_suite)

BOOST_AUTO_TEST_CASE (trademgen_simple_simulation_test) {
    // Input file name
    const stdair::Filename_T lInputFilename (STDAIR_SAMPLE_DIR "/demand01.csv");

```

```

// Generate the date time of the requests with the statistic order method.
const stdair::DemandGenerationMethod lDemandGenerationMethod (stdair::DemandGenerationMethod::STA_ORD);

// State whether the BOM tree should be built-in or parsed from an input file
const bool isBuiltin = false;
BOOST_CHECK_NO_THROW (testDemandGenerationHelper(0,
                                                    lInputFilename,
                                                    lDemandGenerationMethod,
                                                    isBuiltin));
}

BOOST_AUTO_TEST_CASE (trademgen_missing_input_file_test) {

    // Input file name
    const stdair::Filename_T lInputFilename (STDAIR_SAMPLE_DIR "/missingFile.csv");

    // Generate the date time of the requests with the statistic order method.
    const stdair::DemandGenerationMethod lDemandGenerationMethod (stdair::DemandGenerationMethod::STA_ORD);

    // State whether the BOM tree should be built-in or parsed from an input file
    const bool isBuiltin = false;
    BOOST_CHECK_THROW (testDemandGenerationHelper(1,
                                                    lInputFilename,
                                                    lDemandGenerationMethod,
                                                    isBuiltin),
                      TRADEMGEN::DemandInputFileNotFoundException);
}

BOOST_AUTO_TEST_CASE (trademgen_default_bom_simulation_test) {

    // Generate the date time of the requests with the statistic order method.
    const stdair::DemandGenerationMethod lDemandGenerationMethod (stdair::DemandGenerationMethod::STA_ORD);

    // State whether the BOM tree should be built-in or parsed from an input file
    const bool isBuiltin = true;
    BOOST_CHECK_NO_THROW (testDemandGenerationHelper(2,
                                                    " " ,
                                                    lDemandGenerationMethod,
                                                    isBuiltin));
}

BOOST_AUTO_TEST_CASE (trademgen_poisson_process_test) {

    // Generate the date time of the requests with the poisson process.
    const stdair::DemandGenerationMethod lDemandGenerationMethod (stdair::DemandGenerationMethod::POI_PRO);

    // State whether the BOM tree should be built-in or parsed from an input file
    const bool isBuiltin = true;
    BOOST_CHECK_NO_THROW (testDemandGenerationHelper(3,
                                                    " " ,
                                                    lDemandGenerationMethod,
                                                    isBuiltin));
}

// End the test suite
BOOST_AUTO_TEST_SUITE_END()

/*!

```