

# AirTSP Reference Manual

## 1.01.2

Generated by Doxygen 1.4.7

Sat Jun 13 22:39:14 2015

## Contents

|           |                                                       |            |
|-----------|-------------------------------------------------------|------------|
| <b>1</b>  | <b><a href="#">AirTSP Documentation</a></b>           | <b>1</b>   |
| <b>2</b>  | <b><a href="#">AirTSP Directory Hierarchy</a></b>     | <b>2</b>   |
| <b>3</b>  | <b><a href="#">AirTSP Namespace Index</a></b>         | <b>3</b>   |
| <b>4</b>  | <b><a href="#">AirTSP Hierarchical Index</a></b>      | <b>3</b>   |
| <b>5</b>  | <b><a href="#">AirTSP Class Index</a></b>             | <b>8</b>   |
| <b>6</b>  | <b><a href="#">AirTSP File Index</a></b>              | <b>12</b>  |
| <b>7</b>  | <b><a href="#">AirTSP Page Index</a></b>              | <b>15</b>  |
| <b>8</b>  | <b><a href="#">AirTSP Directory Documentation</a></b> | <b>15</b>  |
| <b>9</b>  | <b><a href="#">AirTSP Namespace Documentation</a></b> | <b>18</b>  |
| <b>10</b> | <b><a href="#">AirTSP Class Documentation</a></b>     | <b>35</b>  |
| <b>11</b> | <b><a href="#">AirTSP File Documentation</a></b>      | <b>185</b> |
| <b>12</b> | <b><a href="#">AirTSP Page Documentation</a></b>      | <b>222</b> |

## 1 AirTSP Documentation

### 1.1 Getting Started

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

## 1.2 AirTSP at SourceForge

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

## 1.3 AirTSP 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 AirTSP

## 1.6 About AirTSP

AirTSP is a C++ library of classes and functions modeling airline schedules, for instance allowing to retrieve all the flight-based travel solutions corresponding to a given pair of origin and destination points. AirTSP mainly targets simulation purposes. [N](#)

AirTSP makes an extensive use of existing open-source libraries for increased functionality, speed and accuracy. In particular Boost (*C++ STL Extensions*) library is used.

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

AirTSP 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 AirTSP 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 AirTSP.

## 2 AirTSP Directory Hierarchy

### 2.1 AirTSP Directories

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

|                |           |
|----------------|-----------|
| <b>airtsp</b>  | <b>16</b> |
| <b>basic</b>   | <b>16</b> |
| <b>batches</b> | <b>16</b> |
| <b>bom</b>     | <b>16</b> |
| <b>command</b> | <b>17</b> |
| <b>config</b>  | <b>18</b> |
| <b>factory</b> | <b>18</b> |
| <b>service</b> | <b>18</b> |
| <b>test</b>    | <b>18</b> |
| <b>airtsp</b>  | <b>15</b> |

## 3 AirTSP Namespace Index

### 3.1 AirTSP Namespace List

Here is a list of all namespaces with brief descriptions:

|                                       |           |
|---------------------------------------|-----------|
| <b>airtsp</b>                         | <b>18</b> |
| <b>AIRTSP</b>                         | <b>21</b> |
| <b>AIRTSP::OnDParserHelper</b>        | <b>28</b> |
| <b>AIRTSP::ScheduleParserHelper</b>   | <b>30</b> |
| <b>boost</b> (Forward declarations )  | <b>34</b> |
| <b>boost::serialization</b>           | <b>34</b> |
| <b>boost::spirit::classic</b>         | <b>34</b> |
| <b>stdair</b> (Forward declarations ) | <b>34</b> |

## 4 AirTSP Hierarchical Index

### 4.1 AirTSP Class Hierarchy

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

|                                       |            |
|---------------------------------------|------------|
| <b>airtsp::Airline_T</b>              | <b>35</b>  |
| <b>AIRTSP::AIRTSP_Service</b>         | <b>37</b>  |
| 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>42</b>  |
| <b>AIRTSP::OriginDestinationSet</b>   | <b>85</b>  |
| <b>AIRTSP::ReachableUniverse</b>      | <b>97</b>  |
| <b>AIRTSP::SegmentPathPeriod</b>      | <b>115</b> |
| <b>AIRTSP::BomDisplay</b>             | <b>42</b>  |
| <b>CmdAbstract</b>                    | <b>43</b>  |
| <b>AIRTSP::FlightPeriodFileParser</b> | <b>55</b>  |
| <b>AIRTSP::InventoryGenerator</b>     | <b>69</b>  |
| <b>AIRTSP::OnDParser</b>              | <b>75</b>  |
| <b>AIRTSP::OnDPeriodFileParser</b>    | <b>79</b>  |
| <b>AIRTSP::OnDPeriodGenerator</b>     | <b>80</b>  |
| <b>AIRTSP::ScheduleParser</b>         | <b>104</b> |
| <b>AIRTSP::SegmentPathGenerator</b>   | <b>114</b> |
| <b>AIRTSP::SegmentPathProvider</b>    | <b>125</b> |
| <b>AIRTSP::Simulator</b>              | <b>130</b> |
| <b>AIRTSP::TravelSolutionParser</b>   | <b>184</b> |
| std::complex                          |            |
| <b>TestFixture</b>                    | <b>44</b>  |
| <b>AirlineScheduleTestSuite</b>       | <b>36</b>  |
| <b>airtsp::Date_T</b>                 | <b>45</b>  |

|                                                                                       |            |
|---------------------------------------------------------------------------------------|------------|
| std::deque< T >                                                                       |            |
| 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>AIRTSP::FacServiceAbstract</b>                                                     | <b>51</b>  |
| <b>FacServiceAbstract</b>                                                             | <b>52</b>  |
| <b>AIRTSP::FacAIRTSPServiceContext</b>                                                | <b>49</b>  |
| <b>FileNotFoundException</b>                                                          | <b>54</b>  |
| <b>AIRTSP::OnDInputFileNotFoundException</b>                                          | <b>74</b>  |
| <b>AIRTSP::ScheduleInputFileNotFoundException</b>                                     | <b>104</b> |
| <b>AIRTSP::FlagSaver</b>                                                              | <b>54</b>  |
| <b>AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition&lt; ScannerT &gt;</b> | <b>57</b>  |
| <b>grammar</b>                                                                        | <b>69</b>  |
| <b>AIRTSP::OnDParserHelper::OnDParser</b>                                             | <b>76</b>  |
| <b>AIRTSP::ScheduleParserHelper::FlightPeriodParser</b>                               | <b>56</b>  |
| <b>airtsp::SearchStringParser</b>                                                     | <b>107</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_istream< char >
    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_istream< wchar_t >
        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

```

|                                                                         |            |
|-------------------------------------------------------------------------|------------|
| <b>KeyAbstract</b>                                                      | <b>70</b>  |
| <b>AIRTSP::OriginDestinationSetKey</b>                                  | <b>89</b>  |
| <b>AIRTSP::ReachableUniverseKey</b>                                     | <b>101</b> |
| <b>AIRTSP::SegmentPathPeriodKey</b>                                     | <b>121</b> |
| std::list< T >                                                          |            |
| std::map< K, T >                                                        |            |
| std::multimap< K, T >                                                   |            |
| std::multiset< K >                                                      |            |
| <b>AIRTSP::OnDParserHelper::OnDParser::definition&lt; ScannerT &gt;</b> | <b>77</b>  |
| <b>ParserException</b>                                                  | <b>91</b>  |
| <b>AIRTSP::SegmentDateNotFoundException</b>                             | <b>113</b> |

|                                                                |            |
|----------------------------------------------------------------|------------|
| <b>AIRTSP::OnDParserHelper::ParserSemanticAction</b>           | <b>91</b>  |
| <b>AIRTSP::OnDParserHelper::doEndOnD</b>                       | <b>48</b>  |
| <b>AIRTSP::OnDParserHelper::storeAirlineCode</b>               | <b>139</b> |
| <b>AIRTSP::OnDParserHelper::storeClassCode</b>                 | <b>145</b> |
| <b>AIRTSP::OnDParserHelper::storeDateRangeEnd</b>              | <b>148</b> |
| <b>AIRTSP::OnDParserHelper::storeDateRangeStart</b>            | <b>152</b> |
| <b>AIRTSP::OnDParserHelper::storeDestination</b>               | <b>153</b> |
| <b>AIRTSP::OnDParserHelper::storeEndRangeTime</b>              | <b>158</b> |
| <b>AIRTSP::OnDParserHelper::storeOrigin</b>                    | <b>176</b> |
| <b>AIRTSP::OnDParserHelper::storeStartRangeTime</b>            | <b>183</b> |
| <b>AIRTSP::ScheduleParserHelper::ParserSemanticAction</b>      | <b>93</b>  |
| <b>AIRTSP::ScheduleParserHelper::doEndFlight</b>               | <b>46</b>  |
| <b>AIRTSP::ScheduleParserHelper::storeAirlineCode</b>          | <b>140</b> |
| <b>AIRTSP::ScheduleParserHelper::storeBoardingTime</b>         | <b>142</b> |
| <b>AIRTSP::ScheduleParserHelper::storeCapacity</b>             | <b>143</b> |
| <b>AIRTSP::ScheduleParserHelper::storeClasses</b>              | <b>146</b> |
| <b>AIRTSP::ScheduleParserHelper::storeDateRangeEnd</b>         | <b>149</b> |
| <b>AIRTSP::ScheduleParserHelper::storeDateRangeStart</b>       | <b>150</b> |
| <b>AIRTSP::ScheduleParserHelper::storeDow</b>                  | <b>155</b> |
| <b>AIRTSP::ScheduleParserHelper::storeElapsedTime</b>          | <b>156</b> |
| <b>AIRTSP::ScheduleParserHelper::storeFamilyCode</b>           | <b>159</b> |
| <b>AIRTSP::ScheduleParserHelper::storeFCClasses</b>            | <b>160</b> |
| <b>AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey</b> | <b>162</b> |
| <b>AIRTSP::ScheduleParserHelper::storeFlightNumber</b>         | <b>163</b> |
| <b>AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey</b>        | <b>165</b> |
| <b>AIRTSP::ScheduleParserHelper::storeLegBoardingPoint</b>     | <b>166</b> |
| <b>AIRTSP::ScheduleParserHelper::storeLegCabinCode</b>         | <b>168</b> |
| <b>AIRTSP::ScheduleParserHelper::storeLegOffPoint</b>          | <b>169</b> |
| <b>AIRTSP::ScheduleParserHelper::storeOffTime</b>              | <b>171</b> |



|                                                          |     |
|----------------------------------------------------------|-----|
| AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode  | 173 |
| AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber | 174 |
| AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint  | 177 |
| AIRTSP::ScheduleParserHelper::storeSegmentCabinCode      | 178 |
| AIRTSP::ScheduleParserHelper::storeSegmentOffPoint       | 180 |
| AIRTSP::ScheduleParserHelper::storeSegmentSpecificity    | 181 |
| airtsp::Passenger_T                                      | 94  |
| airtsp::Place_T                                          | 96  |
| std::priority_queue< T >                                 |     |
| std::queue< T >                                          |     |
| airtsp::SearchString_T                                   | 105 |
| airtsp::SearchStringParser::definition< ScannerT >       | 108 |
| AIRTSP::SegmentPeriodHelper                              | 126 |
| AIRTSP::ServiceAbstract                                  | 129 |
| ServiceAbstract                                          | 130 |
| AIRTSP::AIRTSP_ServiceContext                            | 41  |
| std::set< K >                                            |     |
| std::stack< T >                                          |     |
| airtsp::store_adult_passenger_type                       | 131 |
| airtsp::store_airline_code                               | 132 |
| airtsp::store_airline_name                               | 133 |
| airtsp::store_airline_sign                               | 134 |
| airtsp::store_child_passenger_type                       | 135 |
| airtsp::store_date                                       | 136 |
| airtsp::store_passenger_number                           | 136 |
| airtsp::store_pet_passenger_type                         | 137 |
| airtsp::store_place_element                              | 138 |
| StructAbstract                                           | 184 |
| AIRTSP::FareFamilyStruct                                 | 52  |
| AIRTSP::FlightPeriodStruct                               | 62  |
| AIRTSP::LegCabinStruct                                   | 70  |

|                                       |            |
|---------------------------------------|------------|
| <b>AIRTSP::LegStruct</b>              | <b>71</b>  |
| <b>AIRTSP::OnDPeriodStruct</b>        | <b>81</b>  |
| <b>AIRTSP::SegmentCabinStruct</b>     | <b>112</b> |
| <b>AIRTSP::SegmentStruct</b>          | <b>127</b> |
| <code>std::valarray&lt; T &gt;</code> |            |
| <code>std::vector&lt; T &gt;</code>   |            |

## 5 AirTSP Class Index

### 5.1 AirTSP Class List

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

|                                                                                                           |           |
|-----------------------------------------------------------------------------------------------------------|-----------|
| <b><a href="#">airtsp::Airline_T</a></b>                                                                  | <b>35</b> |
| <b><a href="#">AirlineScheduleTestSuite</a></b>                                                           | <b>36</b> |
| <b><a href="#">AIRTSP::AIRTSP_Service</a> (Interface for the Airtsp Services )</b>                        | <b>37</b> |
| <b><a href="#">AIRTSP::AIRTSP_ServiceContext</a> (Class holding the context of the Airtsp services )</b>  | <b>41</b> |
| <b><a href="#">BomAbstract</a></b>                                                                        | <b>42</b> |
| <b><a href="#">AIRTSP::BomDisplay</a> (Utility class to display AirTSP objects with a pretty format )</b> | <b>42</b> |
| <b><a href="#">CmdAbstract</a></b>                                                                        | <b>43</b> |
| <b><a href="#">TestFixture</a></b>                                                                        | <b>44</b> |
| <b><a href="#">airtsp::Date_T</a></b>                                                                     | <b>45</b> |
| <b><a href="#">AIRTSP::ScheduleParserHelper::doEndFlight</a></b>                                          | <b>46</b> |
| <b><a href="#">AIRTSP::OnDParserHelper::doEndOnD</a></b>                                                  | <b>48</b> |
| <b><a href="#">AIRTSP::FacAIRTSPServiceContext</a> (Factory for the service context )</b>                 | <b>49</b> |
| <b><a href="#">AIRTSP::FacServiceAbstract</a></b>                                                         | <b>51</b> |
| <b><a href="#">FacServiceAbstract</a></b>                                                                 | <b>52</b> |
| <b><a href="#">AIRTSP::FareFamilyStruct</a></b>                                                           | <b>52</b> |
| <b><a href="#">FileNotFoundException</a></b>                                                              | <b>54</b> |
| <b><a href="#">AIRTSP::FlagSaver</a></b>                                                                  | <b>54</b> |
| <b><a href="#">AIRTSP::FlightPeriodFileParser</a></b>                                                     | <b>55</b> |
| <b><a href="#">AIRTSP::ScheduleParserHelper::FlightPeriodParser</a></b>                                   | <b>56</b> |
| <b><a href="#">AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition&lt; ScannerT &gt;</a></b>     | <b>57</b> |

|                                                                                                             |            |
|-------------------------------------------------------------------------------------------------------------|------------|
| <b>AIRTSP::FlightPeriodStruct</b>                                                                           | <b>62</b>  |
| <b>grammar</b>                                                                                              | <b>69</b>  |
| <b>AIRTSP::InventoryGenerator</b>                                                                           | <b>69</b>  |
| <b>KeyAbstract</b>                                                                                          | <b>70</b>  |
| <b>AIRTSP::LegCabinStruct</b>                                                                               | <b>70</b>  |
| <b>AIRTSP::LegStruct</b>                                                                                    | <b>71</b>  |
| <b>AIRTSP::OnDInputFileNotFoundException</b>                                                                | <b>74</b>  |
| <b>AIRTSP::OnDParser</b> (Class wrapping the parser entry point )                                           | <b>75</b>  |
| <b>AIRTSP::OnDParserHelper::OnDParser</b>                                                                   | <b>76</b>  |
| <b>AIRTSP::OnDParserHelper::OnDParser::definition&lt; ScannerT &gt;</b>                                     | <b>77</b>  |
| <b>AIRTSP::OnDPeriodFileParser</b>                                                                          | <b>79</b>  |
| <b>AIRTSP::OnDPeriodGenerator</b> (Class handling the generation / instantiation of the O&D-Period BOM )    | <b>80</b>  |
| <b>AIRTSP::OnDPeriodStruct</b>                                                                              | <b>81</b>  |
| <b>AIRTSP::OriginDestinationSet</b> (Class representing a simple sub-network )                              | <b>85</b>  |
| <b>AIRTSP::OriginDestinationSetKey</b> (Structure representing the key of a sub-network )                   | <b>89</b>  |
| <b>ParserException</b>                                                                                      | <b>91</b>  |
| <b>AIRTSP::OnDParserHelper::ParserSemanticAction</b>                                                        | <b>91</b>  |
| <b>AIRTSP::ScheduleParserHelper::ParserSemanticAction</b>                                                   | <b>93</b>  |
| <b>airtsp::Passenger_T</b>                                                                                  | <b>94</b>  |
| <b>airtsp::Place_T</b>                                                                                      | <b>96</b>  |
| <b>AIRTSP::ReachableUniverse</b> (Class representing the root of the schedule-related BOM tree )            | <b>97</b>  |
| <b>AIRTSP::ReachableUniverseKey</b> (Structure representing the key of the schedule-related BOM tree root ) | <b>101</b> |
| <b>AIRTSP::ScheduleInputFileNotFoundException</b>                                                           | <b>104</b> |
| <b>AIRTSP::ScheduleParser</b>                                                                               | <b>104</b> |
| <b>airtsp::SearchString_T</b>                                                                               | <b>105</b> |
| <b>airtsp::SearchStringParser</b>                                                                           | <b>107</b> |
| <b>airtsp::SearchStringParser::definition&lt; ScannerT &gt;</b>                                             | <b>108</b> |
| <b>AIRTSP::SegmentCabinStruct</b>                                                                           | <b>112</b> |

|                                                                                                                               |     |
|-------------------------------------------------------------------------------------------------------------------------------|-----|
| <a href="#">AIRTSP::SegmentDateNotFoundException</a>                                                                          | 113 |
| <a href="#">AIRTSP::SegmentPathGenerator</a> (Class handling the generation / instantiation of the network BOM )              | 114 |
| <a href="#">AIRTSP::SegmentPathPeriod</a> (Class representing a segment/path )                                                | 115 |
| <a href="#">AIRTSP::SegmentPathPeriodKey</a> (Structure representing the key of a segment/path )                              | 121 |
| <a href="#">AIRTSP::SegmentPathProvider</a> (Class building the travel solutions from airline schedules )                     | 125 |
| <a href="#">AIRTSP::SegmentPeriodHelper</a> (Class representing the actual business functions for an airline segment-period ) | 126 |
| <a href="#">AIRTSP::SegmentStruct</a>                                                                                         | 127 |
| <a href="#">AIRTSP::ServiceAbstract</a>                                                                                       | 129 |
| <a href="#">ServiceAbstract</a>                                                                                               | 130 |
| <a href="#">AIRTSP::Simulator</a>                                                                                             | 130 |
| <a href="#">airtsp::store_adult_passenger_type</a>                                                                            | 131 |
| <a href="#">airtsp::store_airline_code</a>                                                                                    | 132 |
| <a href="#">airtsp::store_airline_name</a>                                                                                    | 133 |
| <a href="#">airtsp::store_airline_sign</a>                                                                                    | 134 |
| <a href="#">airtsp::store_child_passenger_type</a>                                                                            | 135 |
| <a href="#">airtsp::store_date</a>                                                                                            | 136 |
| <a href="#">airtsp::store_passenger_number</a>                                                                                | 136 |
| <a href="#">airtsp::store_pet_passenger_type</a>                                                                              | 137 |
| <a href="#">airtsp::store_place_element</a>                                                                                   | 138 |
| <a href="#">AIRTSP::OnDParserHelper::storeAirlineCode</a>                                                                     | 139 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeAirlineCode</a>                                                                | 140 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeBoardingTime</a>                                                               | 142 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeCapacity</a>                                                                   | 143 |
| <a href="#">AIRTSP::OnDParserHelper::storeClassCode</a>                                                                       | 145 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeClasses</a>                                                                    | 146 |
| <a href="#">AIRTSP::OnDParserHelper::storeDateRangeEnd</a>                                                                    | 148 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeDateRangeEnd</a>                                                               | 149 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeDateRangeStart</a>                                                             | 150 |

|                                                                                                                                                                           |     |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| <a href="#">AIRTSP::OnDParserHelper::storeDateRangeStart</a>                                                                                                              | 152 |
| <a href="#">AIRTSP::OnDParserHelper::storeDestination</a>                                                                                                                 | 153 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeDow</a>                                                                                                                    | 155 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeElapsedTime</a>                                                                                                            | 156 |
| <a href="#">AIRTSP::OnDParserHelper::storeEndRangeTime</a>                                                                                                                | 158 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeFamilyCode</a>                                                                                                             | 159 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeFClasses</a>                                                                                                               | 160 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey</a>                                                                                                   | 162 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeFlightNumber</a>                                                                                                           | 163 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey</a>                                                                                                          | 165 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeLegBoardingPoint</a>                                                                                                       | 166 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeLegCabinCode</a>                                                                                                           | 168 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeLegOffPoint</a>                                                                                                            | 169 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeOffTime</a>                                                                                                                | 171 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode</a>                                                                                                   | 173 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber</a>                                                                                                  | 174 |
| <a href="#">AIRTSP::OnDParserHelper::storeOrigin</a>                                                                                                                      | 176 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint</a>                                                                                                   | 177 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeSegmentCabinCode</a>                                                                                                       | 178 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeSegmentOffPoint</a>                                                                                                        | 180 |
| <a href="#">AIRTSP::ScheduleParserHelper::storeSegmentSpecificity</a>                                                                                                     | 181 |
| <a href="#">AIRTSP::OnDParserHelper::storeStartRangeTime</a>                                                                                                              | 183 |
| <a href="#">StructAbstract</a>                                                                                                                                            | 184 |
| <a href="#">AIRTSP::TravelSolutionParser</a> (Class filling the TravelSolutionHolder structure (representing a list of classes/travelSolutions) from a given input file ) | 184 |

## 6 AirTSP File Index

### 6.1 AirTSP File List

Here is a list of all files with brief descriptions:

|                                           |     |
|-------------------------------------------|-----|
| <a href="#">airtsp/AIRTSP_Service.hpp</a> | 185 |
|-------------------------------------------|-----|

|                                                          |     |
|----------------------------------------------------------|-----|
| <a href="#">airtsp/AIRTSP_Types.hpp</a>                  | 186 |
| <a href="#">airtsp/basic/BasConst.cpp</a>                | 186 |
| <a href="#">airtsp/basic/BasConst_AIRTSP_Service.hpp</a> | 187 |
| <a href="#">airtsp/basic/BasConst_General.hpp</a>        | 187 |
| <a href="#">airtsp/basic/BasParserTypes.hpp</a>          | 187 |
| <a href="#">airtsp/batches/airtsp.cpp</a>                | 188 |
| <a href="#">airtsp/batches/BookingRequestParser.cpp</a>  | 191 |
| <a href="#">airtsp/batches/BookingRequestParser.hpp</a>  | 192 |
| <a href="#">airtsp/bom/AirportList.hpp</a>               | 193 |
| <a href="#">airtsp/bom/BomDisplay.cpp</a>                | 193 |
| <a href="#">airtsp/bom/BomDisplay.hpp</a>                | 194 |
| <a href="#">airtsp/bom/FareFamilyStruct.cpp</a>          | 194 |
| <a href="#">airtsp/bom/FareFamilyStruct.hpp</a>          | 194 |
| <a href="#">airtsp/bom/FlightPeriodStruct.cpp</a>        | 195 |
| <a href="#">airtsp/bom/FlightPeriodStruct.hpp</a>        | 195 |
| <a href="#">airtsp/bom/LegCabinStruct.cpp</a>            | 196 |
| <a href="#">airtsp/bom/LegCabinStruct.hpp</a>            | 196 |
| <a href="#">airtsp/bom/LegStruct.cpp</a>                 | 196 |
| <a href="#">airtsp/bom/LegStruct.hpp</a>                 | 197 |
| <a href="#">airtsp/bom/OnDPeriodStruct.cpp</a>           | 197 |
| <a href="#">airtsp/bom/OnDPeriodStruct.hpp</a>           | 197 |
| <a href="#">airtsp/bom/OriginDestinationSet.cpp</a>      | 198 |
| <a href="#">airtsp/bom/OriginDestinationSet.hpp</a>      | 198 |
| <a href="#">airtsp/bom/OriginDestinationSetKey.cpp</a>   | 199 |
| <a href="#">airtsp/bom/OriginDestinationSetKey.hpp</a>   | 199 |
| <a href="#">airtsp/bom/OriginDestinationSetTypes.hpp</a> | 200 |
| <a href="#">airtsp/bom/ReachableUniverse.cpp</a>         | 200 |
| <a href="#">airtsp/bom/ReachableUniverse.hpp</a>         | 200 |
| <a href="#">airtsp/bom/ReachableUniverseKey.cpp</a>      | 201 |

|                                                         |     |
|---------------------------------------------------------|-----|
| <a href="#">airtsp/bom/ReachableUniverseKey.hpp</a>     | 201 |
| <a href="#">airtsp/bom/ReachableUniverseTypes.hpp</a>   | 202 |
| <a href="#">airtsp/bom/SegmentCabinStruct.cpp</a>       | 202 |
| <a href="#">airtsp/bom/SegmentCabinStruct.hpp</a>       | 202 |
| <a href="#">airtsp/bom/SegmentPathPeriod.cpp</a>        | 203 |
| <a href="#">airtsp/bom/SegmentPathPeriod.hpp</a>        | 204 |
| <a href="#">airtsp/bom/SegmentPathPeriodKey.cpp</a>     | 204 |
| <a href="#">airtsp/bom/SegmentPathPeriodKey.hpp</a>     | 205 |
| <a href="#">airtsp/bom/SegmentPathPeriodTypes.hpp</a>   | 205 |
| <a href="#">airtsp/bom/SegmentPeriodHelper.cpp</a>      | 206 |
| <a href="#">airtsp/bom/SegmentPeriodHelper.hpp</a>      | 206 |
| <a href="#">airtsp/bom/SegmentStruct.cpp</a>            | 206 |
| <a href="#">airtsp/bom/SegmentStruct.hpp</a>            | 206 |
| <a href="#">airtsp/command/InventoryGenerator.cpp</a>   | 207 |
| <a href="#">airtsp/command/InventoryGenerator.hpp</a>   | 207 |
| <a href="#">airtsp/command/OnDParser.cpp</a>            | 208 |
| <a href="#">airtsp/command/OnDParser.hpp</a>            | 208 |
| <a href="#">airtsp/command/OnDParserHelper.cpp</a>      | 209 |
| <a href="#">airtsp/command/OnDParserHelper.hpp</a>      | 209 |
| <a href="#">airtsp/command/OnDPeriodGenerator.cpp</a>   | 210 |
| <a href="#">airtsp/command/OnDPeriodGenerator.hpp</a>   | 210 |
| <a href="#">airtsp/command/ScheduleParser.cpp</a>       | 211 |
| <a href="#">airtsp/command/ScheduleParser.hpp</a>       | 211 |
| <a href="#">airtsp/command/ScheduleParserHelper.cpp</a> | 211 |
| <a href="#">airtsp/command/ScheduleParserHelper.hpp</a> | 212 |
| <a href="#">airtsp/command/SegmentPathGenerator.cpp</a> | 213 |
| <a href="#">airtsp/command/SegmentPathGenerator.hpp</a> | 214 |
| <a href="#">airtsp/command/SegmentPathProvider.cpp</a>  | 214 |
| <a href="#">airtsp/command/SegmentPathProvider.hpp</a>  | 215 |

|                                                            |     |
|------------------------------------------------------------|-----|
| <a href="#">airtsp/command/Simulator.cpp</a>               | 215 |
| <a href="#">airtsp/command/Simulator.hpp</a>               | 215 |
| <a href="#">airtsp/command/TravelSolutionParser.cpp</a>    | 216 |
| <a href="#">airtsp/command/TravelSolutionParser.hpp</a>    | 216 |
| <a href="#">airtsp/config/airtsp-paths.hpp.in</a>          | 216 |
| <a href="#">airtsp/factory/FacAIRTSPServiceContext.cpp</a> | 216 |
| <a href="#">airtsp/factory/FacAIRTSPServiceContext.hpp</a> | 217 |
| <a href="#">airtsp/factory/FacServiceAbstract.cpp</a>      | 217 |
| <a href="#">airtsp/factory/FacServiceAbstract.hpp</a>      | 217 |
| <a href="#">airtsp/service/AIRTSP_Service.cpp</a>          | 218 |
| <a href="#">airtsp/service/AIRTSP_ServiceContext.cpp</a>   | 218 |
| <a href="#">airtsp/service/AIRTSP_ServiceContext.hpp</a>   | 218 |
| <a href="#">airtsp/service/ServiceAbstract.cpp</a>         | 219 |
| <a href="#">airtsp/service/ServiceAbstract.hpp</a>         | 219 |
| <a href="#">test/airtsp/AirlineScheduleTestSuite.cpp</a>   | 221 |
| <a href="#">test/airtsp/AirlineScheduleTestSuite.hpp</a>   | 221 |

## 7 AirTSP Page Index

### 7.1 AirTSP Related Pages

Here is a list of all related documentation pages:

|                                                          |     |
|----------------------------------------------------------|-----|
| <a href="#">Configuration helper for AirTSP programs</a> | 222 |
| <a href="#">People</a>                                   | 222 |
| <a href="#">Coding Rules</a>                             | 223 |
| <a href="#">Copyright and License</a>                    | 224 |
| <a href="#">Documentation Rules</a>                      | 230 |
| <a href="#">Main features</a>                            | 232 |
| <a href="#">Make a Difference</a>                        | 232 |
| <a href="#">Make a new release</a>                       | 233 |
| <a href="#">Installation</a>                             | 236 |



|                                                                        |            |
|------------------------------------------------------------------------|------------|
| <b>Linking with AirTSP</b>                                             | <b>245</b> |
| <b>Test Rules</b>                                                      | <b>247</b> |
| <b>Users Guide</b>                                                     | <b>248</b> |
| <b>Supported Systems</b>                                               | <b>299</b> |
| <b>AirTSP Supported Systems (Previous Releases)</b>                    | <b>308</b> |
| <b>Tutorials</b>                                                       | <b>308</b> |
| <b>Command-Line Test to Demonstrate How To Test the AirTSP Project</b> | <b>311</b> |

## 8 AirTSP Directory Documentation

### 8.1 test/airtsp/ Directory Reference

#### Files

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

### 8.2 airtsp/ Directory Reference

#### Directories

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

#### Files

- file [AIRTSP\\_Service.hpp](#)
- file [AIRTSP\\_Types.hpp](#)

### 8.3 airtsp/basic/ Directory Reference

#### Files

- file [BasConst.cpp](#)
- file [BasConst\\_AIRTSP\\_Service.hpp](#)
- file [BasConst\\_General.hpp](#)
- file [BasParserTypes.hpp](#)

## 8.4 airtsp/batches/ Directory Reference

### Files

- file [airtsp.cpp](#)
- file [BookingRequestParser.cpp](#)
- file [BookingRequestParser.hpp](#)

## 8.5 airtsp/bom/ Directory Reference

### Files

- file [AirportList.hpp](#)
- file [BomDisplay.cpp](#)
- file [BomDisplay.hpp](#)
- file [FareFamilyStruct.cpp](#)
- file [FareFamilyStruct.hpp](#)
- file [FlightPeriodStruct.cpp](#)
- file [FlightPeriodStruct.hpp](#)
- file [LegCabinStruct.cpp](#)
- file [LegCabinStruct.hpp](#)
- file [LegStruct.cpp](#)
- file [LegStruct.hpp](#)
- file [OnDPeriodStruct.cpp](#)
- file [OnDPeriodStruct.hpp](#)
- file [OriginDestinationSet.cpp](#)
- file [OriginDestinationSet.hpp](#)
- file [OriginDestinationSetKey.cpp](#)
- file [OriginDestinationSetKey.hpp](#)
- file [OriginDestinationSetTypes.hpp](#)
- file [ReachableUniverse.cpp](#)
- file [ReachableUniverse.hpp](#)
- file [ReachableUniverseKey.cpp](#)
- file [ReachableUniverseKey.hpp](#)
- file [ReachableUniverseTypes.hpp](#)
- file [SegmentCabinStruct.cpp](#)
- file [SegmentCabinStruct.hpp](#)
- file [SegmentPathPeriod.cpp](#)
- file [SegmentPathPeriod.hpp](#)
- file [SegmentPathPeriodKey.cpp](#)
- file [SegmentPathPeriodKey.hpp](#)
- file [SegmentPathPeriodTypes.hpp](#)
- file [SegmentPeriodHelper.cpp](#)
- file [SegmentPeriodHelper.hpp](#)
- file [SegmentStruct.cpp](#)
- file [SegmentStruct.hpp](#)

## 8.6 airtsp/command/ Directory Reference

### Files

- file [InventoryGenerator.cpp](#)
- file [InventoryGenerator.hpp](#)
- file [OnDParser.cpp](#)
- file [OnDParser.hpp](#)
- file [OnDParserHelper.cpp](#)
- file [OnDParserHelper.hpp](#)
- file [OnDPeriodGenerator.cpp](#)
- file [OnDPeriodGenerator.hpp](#)
- file [ScheduleParser.cpp](#)
- file [ScheduleParser.hpp](#)
- file [ScheduleParserHelper.cpp](#)
- file [ScheduleParserHelper.hpp](#)
- file [SegmentPathGenerator.cpp](#)
- file [SegmentPathGenerator.hpp](#)
- file [SegmentPathProvider.cpp](#)
- file [SegmentPathProvider.hpp](#)
- file [Simulator.cpp](#)
- file [Simulator.hpp](#)
- file [TravelSolutionParser.cpp](#)
- file [TravelSolutionParser.hpp](#)

## 8.7 airtsp/config/ Directory Reference

### Files

- file [airtsp-paths.hpp.in](#)

## 8.8 airtsp/factory/ Directory Reference

### Files

- file [FacAIRTSPServiceContext.cpp](#)
- file [FacAIRTSPServiceContext.hpp](#)
- file [FacServiceAbstract.cpp](#)
- file [FacServiceAbstract.hpp](#)

## 8.9 airtsp/service/ Directory Reference

### Files

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

## 8.10 test/ Directory Reference

### Directories

- directory [airtsp](#)

## 9 AirTSP Namespace Documentation

### 9.1 airtsp Namespace Reference

#### Classes

- struct [store\\_place\\_element](#)
- struct [store\\_date](#)
- struct [store\\_airline\\_sign](#)
- struct [store\\_airline\\_code](#)
- struct [store\\_airline\\_name](#)
- struct [store\\_passenger\\_number](#)
- struct [store\\_adult\\_passenger\\_type](#)
- struct [store\\_child\\_passenger\\_type](#)
- struct [store\\_pet\\_passenger\\_type](#)
- struct [SearchStringParser](#)
- struct [Place\\_T](#)
- struct [Date\\_T](#)
- struct [Airline\\_T](#)
- struct [Passenger\\_T](#)
- struct [SearchString\\_T](#)

#### Typedefs

- typedef std::vector< [Place\\_T](#) > [PlaceList\\_T](#)
- typedef std::vector< [Date\\_T](#) > [DateList\\_T](#)
- typedef std::vector< [Airline\\_T](#) > [AirlineList\\_T](#)
- typedef std::vector< [Passenger\\_T](#) > [PassengerList\\_T](#)

#### Functions

- [SearchString\\_T](#) [parseBookingRequest](#) (const std::string &iSearchString)

#### Variables

- boost::spirit::classic::int\_parser< unsigned int, 10, 1, 1 > [int1\\_p](#)
- boost::spirit::classic::uint\_parser< unsigned int, 10, 1, 1 > [uint1\\_p](#)
- boost::spirit::classic::uint\_parser< unsigned int, 10, 1, 2 > [uint1\\_2\\_p](#)
- boost::spirit::classic::uint\_parser< int, 10, 2, 2 > [uint2\\_p](#)
- boost::spirit::classic::uint\_parser< int, 10, 2, 4 > [uint2\\_4\\_p](#)
- boost::spirit::classic::uint\_parser< int, 10, 4, 4 > [uint4\\_p](#)
- boost::spirit::classic::uint\_parser< int, 10, 1, 4 > [uint1\\_4\\_p](#)

### 9.1.1 Typedef Documentation

#### 9.1.1.1 `typedef std::vector<Place_T> airtsp::PlaceList_T`

List of Place strucutres.

Definition at line 24 of file BookingRequestParser.hpp.

#### 9.1.1.2 `typedef std::vector<Date_T> airtsp::DateList_T`

List of Date strucutres.

Definition at line 49 of file BookingRequestParser.hpp.

#### 9.1.1.3 `typedef std::vector<Airline_T> airtsp::AirlineList_T`

List of Airline strucutres.

Definition at line 68 of file BookingRequestParser.hpp.

#### 9.1.1.4 `typedef std::vector<Passenger_T> airtsp::PassengerList_T`

List of Passenger strucutres.

Definition at line 91 of file BookingRequestParser.hpp.

### 9.1.2 Function Documentation

#### 9.1.2.1 `SearchString_T airtsp::parseBookingRequest (const std::string & iSearchString)`

Parse the booking request.

Sample guadeloupe rio de janeiro 07/22/2009 +aa -ua 2 adults 1 dog

Grammar: search\_string ::= places [dates] (preferred\_airlines) (passengers) dates ::= board\_date [off\_date]  
 places ::= [board\_place] off\_place board\_place ::= place\_elements off\_place ::= place\_elements place\_  
 elements ::= country | city | airport country ::= country\_code | country\_name city ::= city\_code | city\_name  
 airport ::= airport\_code | airport\_name preferred\_airlines ::= [+|-] airline\_code | airline\_name passen-  
 gers ::= adult\_number adult\_description [child\_number child\_description] [pet\_number pet\_description]  
 adult\_description ::= 'adult' | 'adults' | 'pax' | 'passengers' child\_description ::= 'child' | 'children' | 'kid'  
 | 'kids' pet\_description ::= 'dog' | 'dogs' | 'cat' | 'cats'

Definition at line 373 of file BookingRequestParser.cpp.

### 9.1.3 Variable Documentation

#### 9.1.3.1 `boost::spirit::classic::int_parser<unsigned int, 10, 1, 1> airtsp::int1_p`

1-digit-integer parser

Definition at line 203 of file BookingRequestParser.cpp.

#### 9.1.3.2 `boost::spirit::classic::uint_parser<unsigned int, 10, 1, 1> airtsp::uint1_p`

1-digit-integer parser

Definition at line 205 of file BookingRequestParser.cpp.

Referenced by `airtsp::SearchStringParser::definition< ScannerT >::definition()`.

**9.1.3.3 `boost::spirit::classic::uint_parser<unsigned int, 10, 1, 2>` [airtsp::uint1\\_2\\_p](#)**

Up-to-2-digit-integer parser

Definition at line 207 of file BookingRequestParser.cpp.

Referenced by `airtsp::SearchStringParser::definition< ScannerT >::definition()`.

**9.1.3.4 `boost::spirit::classic::uint_parser<int, 10, 2, 2>` [airtsp::uint2\\_p](#)**

2-digit-integer parser

Definition at line 209 of file BookingRequestParser.cpp.

Referenced by `airtsp::SearchStringParser::definition< ScannerT >::definition()`.

**9.1.3.5 `boost::spirit::classic::uint_parser<int, 10, 2, 4>` [airtsp::uint2\\_4\\_p](#)**

Up-to-4-digit-integer parser

Definition at line 211 of file BookingRequestParser.cpp.

**9.1.3.6 `boost::spirit::classic::uint_parser<int, 10, 4, 4>` [airtsp::uint4\\_p](#)**

4-digit-integer parser

Definition at line 213 of file BookingRequestParser.cpp.

Referenced by `airtsp::SearchStringParser::definition< ScannerT >::definition()`.

**9.1.3.7 `boost::spirit::classic::uint_parser<int, 10, 1, 4>` [airtsp::uint1\\_4\\_p](#)**

Up-to-4-digit-integer parser

Definition at line 215 of file BookingRequestParser.cpp.

**9.2 AIRTSP Namespace Reference****Classes**

- class [AIRTSP\\_Service](#)  
*Interface for the Airtsp Services.*
- class [SegmentDateNotFoundException](#)
- class [OnDInputFileNotFoundException](#)
- class [ScheduleInputFileNotFoundException](#)
- struct [FlagSaver](#)
- class [BomDisplay](#)  
*Utility class to display AirTSP objects with a pretty format.*
- struct [FareFamilyStruct](#)
- struct [FlightPeriodStruct](#)
- struct [LegCabinStruct](#)
- struct [LegStruct](#)
- struct [OnDPeriodStruct](#)
- class [OriginDestinationSet](#)

*Class representing a simple sub-network.*

- struct [OriginDestinationSetKey](#)

*Structure representing the key of a sub-network.*

- class [ReachableUniverse](#)

*Class representing the root of the schedule-related BOM tree.*

- struct [ReachableUniverseKey](#)

*Structure representing the key of the schedule-related BOM tree root.*

- struct [SegmentCabinStruct](#)

- class [SegmentPathPeriod](#)

*Class representing a segment/path.*

- struct [SegmentPathPeriodKey](#)

*Structure representing the key of a segment/path.*

- class [SegmentPeriodHelper](#)

*Class representing the actual business functions for an airline segment-period.*

- struct [SegmentStruct](#)

- class [InventoryGenerator](#)

- class [OnDParser](#)

*Class wrapping the parser entry point.*

- class [OnDPeriodFileParser](#)

- class [OnDPeriodGenerator](#)

*Class handling the generation / instantiation of the O&D-Period BOM.*

- class [ScheduleParser](#)

- class [FlightPeriodFileParser](#)

- class [SegmentPathGenerator](#)

*Class handling the generation / instantiation of the network BOM.*

- class [SegmentPathProvider](#)

*Class building the travel solutions from airline schedules.*

- class [Simulator](#)

- class [TravelSolutionParser](#)

*Class filling the TravelSolutionHolder structure (representing a list of classes/travelSolutions) from a given input file.*

- class [FacAIRTSPServiceContext](#)

*Factory for the service context.*

- class [FacServiceAbstract](#)

- class [AIRTSP\\_ServiceContext](#)

*Class holding the context of the Airtsp services.*

- class [ServiceAbstract](#)

## Namespaces

- namespace [ScheduleParserHelper](#)
- namespace [OnDParserHelper](#)

## Typedefs

- typedef boost::shared\_ptr< [AIRTSP\\_Service](#) > [AIRTSP\\_ServicePtr\\_T](#)
- 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, 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< [uint4\\_p\\_t](#), unsigned int > [bounded4\\_p\\_t](#)
- typedef boost::spirit::classic::bounded< [uint1\\_4\\_p\\_t](#), unsigned int > [bounded1\\_4\\_p\\_t](#)
- typedef std::set< stdair::AirportCode\_T > [AirportList\\_T](#)
- typedef std::vector< stdair::AirportCode\_T > [AirportOrderedList\\_T](#)
- typedef std::vector< [FareFamilyStruct](#) > [FareFamilyStructList\\_T](#)
- typedef std::vector< [LegCabinStruct](#) > [LegCabinStructList\\_T](#)
- typedef std::vector< [LegStruct](#) > [LegStructList\\_T](#)
- typedef std::list< [OriginDestinationSet](#) \* > [OriginDestinationSetList\\_T](#)
- typedef std::map< const stdair::MapKey\_T, [OriginDestinationSet](#) \* > [OriginDestinationSetMap\\_T](#)
- typedef std::list< [ReachableUniverse](#) \* > [ReachableUniverseList\\_T](#)
- typedef std::map< const stdair::MapKey\_T, [ReachableUniverse](#) \* > [ReachableUniverseMap\\_T](#)
- typedef std::vector< [SegmentCabinStruct](#) > [SegmentCabinStructList\\_T](#)
- typedef std::list< [SegmentPathPeriod](#) \* > [SegmentPathPeriodList\\_T](#)
- typedef std::multimap< const stdair::MapKey\_T, [SegmentPathPeriod](#) \* > [SegmentPathPeriod-Multimap\\_T](#)
- typedef std::vector< const [SegmentPathPeriod](#) \* > [SegmentPathPeriodLightList\\_T](#)
- typedef std::vector< [SegmentPathPeriodLightList\\_T](#) > [SegmentPathPeriodListList\\_T](#)
- typedef std::vector< stdair::DateOffset\_T > [DateOffsetList\\_T](#)
- typedef std::vector< [SegmentStruct](#) > [SegmentStructList\\_T](#)

## Functions

- const stdair::Duration\_T [MINIMUM\\_TIME\\_BETWEEN\\_REQUEST\\_AND\\_DEPARTURE](#) (4, 0)
- template void [OriginDestinationSet::serialize](#)< ba::text\_oarchive > (ba::text\_oarchive &, unsigned int)
- template void [OriginDestinationSet::serialize](#)< ba::text\_iarchive > (ba::text\_iarchive &, unsigned int)
- template void [OriginDestinationSetKey::serialize](#)< ba::text\_oarchive > (ba::text\_oarchive &, unsigned int)



- template void [OriginDestinationSetKey::serialize](#)< [ba::text\\_iarchive](#) > ([ba::text\\_iarchive](#) &, unsigned int)
- template void [ReachableUniverse::serialize](#)< [ba::text\\_oarchive](#) > ([ba::text\\_oarchive](#) &, unsigned int)
- template void [ReachableUniverse::serialize](#)< [ba::text\\_iarchive](#) > ([ba::text\\_iarchive](#) &, unsigned int)
- template void [ReachableUniverseKey::serialize](#)< [ba::text\\_oarchive](#) > ([ba::text\\_oarchive](#) &, unsigned int)
- template void [ReachableUniverseKey::serialize](#)< [ba::text\\_iarchive](#) > ([ba::text\\_iarchive](#) &, unsigned int)
- template void [SegmentPathPeriod::serialize](#)< [ba::text\\_oarchive](#) > ([ba::text\\_oarchive](#) &, unsigned int)
- template void [SegmentPathPeriod::serialize](#)< [ba::text\\_iarchive](#) > ([ba::text\\_iarchive](#) &, unsigned int)
- template void [SegmentPathPeriodKey::serialize](#)< [ba::text\\_oarchive](#) > ([ba::text\\_oarchive](#) &, unsigned int)
- template void [SegmentPathPeriodKey::serialize](#)< [ba::text\\_iarchive](#) > ([ba::text\\_iarchive](#) &, unsigned int)

## Variables

- const int [DEFAULT\\_NUMBER\\_OF\\_DRAWS\\_FOR\\_MC\\_SIMULATION](#) = 100000
- const int [DEFAULT\\_NUMBER\\_OF\\_DRAWS\\_FOR\\_MC\\_SIMULATION](#)
- const std::duration<T> [MINIMUM\\_TIME\\_BETWEEN\\_REQUEST\\_AND\\_DEPARTURE](#)

### 9.2.1 Typedef Documentation

#### 9.2.1.1 typedef boost::shared\_ptr<[AIRTSP\\_Service](#)> [AIRTSP::AIRTSP\\_ServicePtr\\_T](#)

(Smart) Pointer on the Airtsp service handler.

Definition at line 62 of file [AIRTSP\\_Types.hpp](#).

#### 9.2.1.2 typedef char [AIRTSP::char\\_t](#)

Definition at line 31 of file [BasParserTypes.hpp](#).

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

Definition at line 35 of file [BasParserTypes.hpp](#).

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

Definition at line 36 of file [BasParserTypes.hpp](#).

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

Definition at line 37 of file [BasParserTypes.hpp](#).

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

1-digit-integer parser

Definition at line 45 of file [BasParserTypes.hpp](#).

**9.2.1.7** `typedef boost::spirit::classic::uint_parser<unsigned int, 10, 2, 2> AIRTSP::uint2_p_t`

2-digit-integer parser

Definition at line 48 of file BasParserTypes.hpp.

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

4-digit-integer parser

Definition at line 51 of file BasParserTypes.hpp.

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

Up-to-4-digit-integer parser

Definition at line 54 of file BasParserTypes.hpp.

**9.2.1.10** `typedef boost::spirit::classic::chset<char_t> AIRTSP::chset_t`

character set

Definition at line 57 of file BasParserTypes.hpp.

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

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

Definition at line 63 of file BasParserTypes.hpp.

**9.2.1.12** `typedef boost::spirit::classic::bounded<uint2_p_t, unsigned int> AIRTSP::bounded2_p_t`

Bounded-number-of-integers parser

Definition at line 66 of file BasParserTypes.hpp.

**9.2.1.13** `typedef boost::spirit::classic::bounded<uint4_p_t, unsigned int> AIRTSP::bounded4_p_t`

Definition at line 67 of file BasParserTypes.hpp.

**9.2.1.14** `typedef boost::spirit::classic::bounded<uint1_4_p_t, unsigned int> AIRTSP::bounded1_4_p_t`

Definition at line 68 of file BasParserTypes.hpp.

**9.2.1.15** `typedef std::set<stdair::AirportCode_T> AIRTSP::AirportList_T`

Define lists of Airport Codes.

Definition at line 16 of file AirportList.hpp.

**9.2.1.16** `typedef std::vector<stdair::AirportCode_T> AIRTSP::AirportOrderedList_T`

Definition at line 17 of file AirportList.hpp.

**9.2.1.17** `typedef std::vector<FareFamilyStruct> AIRTSP::FareFamilyStructList_T`

List of FareFamily-Detail structures.

Definition at line 34 of file FareFamilyStruct.hpp.

**9.2.1.18** `typedef std::vector<LegCabinStruct> AIRTSP::LegCabinStructList_T`

List of LegCabin-Detail structures.

Definition at line 36 of file LegCabinStruct.hpp.

**9.2.1.19** `typedef std::vector<LegStruct> AIRTSP::LegStructList_T`

List of Leg structures.

Definition at line 52 of file LegStruct.hpp.

**9.2.1.20** `typedef std::list<OriginDestinationSet\*> AIRTSP::OriginDestinationSetList_T`

Define the [OriginDestinationSet](#) list.

Definition at line 18 of file OriginDestinationSetTypes.hpp.

**9.2.1.21** `typedef std::map<const stdair::MapKey_T, OriginDestinationSet\*> AIRTSP::OriginDestinationSetMap_T`

Define the [OriginDestinationSet](#) map.

Definition at line 25 of file OriginDestinationSetTypes.hpp.

**9.2.1.22** `typedef std::list<ReachableUniverse\*> AIRTSP::ReachableUniverseList_T`

Define the reachable-universe list.

Definition at line 18 of file ReachableUniverseTypes.hpp.

**9.2.1.23** `typedef std::map<const stdair::MapKey_T, ReachableUniverse\*> AIRTSP::ReachableUniverseMap_T`

Define the reachable-universe map.

Definition at line 25 of file ReachableUniverseTypes.hpp.

**9.2.1.24** `typedef std::vector<SegmentCabinStruct> AIRTSP::SegmentCabinStructList_T`

List of SegmentCabin-Detail structures.

Definition at line 43 of file SegmentCabinStruct.hpp.

**9.2.1.25** `typedef std::list<SegmentPathPeriod\*> AIRTSP::SegmentPathPeriodList_T`

Define the segment path list.

Definition at line 20 of file SegmentPathPeriodTypes.hpp.

**9.2.1.26** `typedef std::multimap<const stdair::MapKey_T, SegmentPathPeriod*> AIRTSP::SegmentPathPeriodMultimap_T`

Define the segment path map.

Definition at line 27 of file SegmentPathPeriodTypes.hpp.

**9.2.1.27** `typedef std::vector<const SegmentPathPeriod*> AIRTSP::SegmentPathPeriodLightList_T`

Define the (temporary) containers of segment path period.

Definition at line 30 of file SegmentPathPeriodTypes.hpp.

**9.2.1.28** `typedef std::vector<SegmentPathPeriodLightList_T> AIRTSP::SegmentPathPeriodListList_T`

Definition at line 31 of file SegmentPathPeriodTypes.hpp.

**9.2.1.29** `typedef std::vector<stdair::DateOffset_T> AIRTSP::DateOffsetList_T`

Define the vector of boarding date offsets of the member segments of a segment path compare to the boarding date of the first segment.

Definition at line 35 of file SegmentPathPeriodTypes.hpp.

**9.2.1.30** `typedef std::vector<SegmentStruct> AIRTSP::SegmentStructList_T`

List of Segment strucutres.

Definition at line 44 of file SegmentStruct.hpp.

## 9.2.2 Function Documentation

**9.2.2.1** `const stdair::Duration_T AIRTSP::MINIMUM_TIME_BETWEEN_REQUEST_AND_DEPARTURE (4, 0, 0)`

Default value for the minimum time between the request and the departure time.

**9.2.2.2** `template void AIRTSP::OriginDestinationSet::serialize< ba::text_oarchive > (ba::text_oarchive &, unsigned int)`

**9.2.2.3** `template void AIRTSP::OriginDestinationSet::serialize< ba::text_iarchive > (ba::text_iarchive &, unsigned int)`

**9.2.2.4** `template void AIRTSP::OriginDestinationSetKey::serialize< ba::text_oarchive > (ba::text_oarchive &, unsigned int)`

**9.2.2.5** `template void AIRTSP::OriginDestinationSetKey::serialize< ba::text_iarchive > (ba::text_iarchive &, unsigned int)`

**9.2.2.6** `template void AIRTSP::ReachableUniverse::serialize< ba::text_oarchive > (ba::text_oarchive &, unsigned int)`

**9.2.2.7** `template void AIRTSP::ReachableUniverse::serialize< ba::text_iarchive > (ba::text_iarchive &, unsigned int)`

**9.2.2.8** `template void AIRTSP::ReachableUniverseKey::serialize< ba::text_oarchive > (ba::text_oarchive &, unsigned int)`

**9.2.2.9** `template void AIRTSP::ReachableUniverseKey::serialize< ba::text_iarchive > (ba::text_iarchive &, unsigned int)`

**9.2.2.10** `template void AIRTSP::SegmentPathPeriod::serialize< ba::text_oarchive > (ba::text_oarchive &, unsigned int)`

**9.2.2.11** `template void AIRTSP::SegmentPathPeriod::serialize< ba::text_iarchive > (ba::text_iarchive &, unsigned int)`

**9.2.2.12** `template void AIRTSP::SegmentPathPeriodKey::serialize< ba::text_oarchive > (ba::text_oarchive &, unsigned int)`

**9.2.2.13** `template void AIRTSP::SegmentPathPeriodKey::serialize< ba::text_iarchive > (ba::text_iarchive &, unsigned int)`

### 9.2.3 Variable Documentation

**9.2.3.1** `const int AIRTSP::DEFAULT_NUMBER_OF_DRAWS_FOR_MC_SIMULATION = 100000`

Default value for the number of draws within the Monte-Carlo Integration algorithm.

Definition at line 8 of file BasConst.cpp.

**9.2.3.2** `const int AIRTSP::DEFAULT_NUMBER_OF_DRAWS_FOR_MC_SIMULATION`

Default value for the number of draws within the Monte-Carlo Integration algorithm.

Definition at line 8 of file BasConst.cpp.

**9.2.3.3** `const stdair::Duration_T AIRTSP::MINIMUM_TIME_BETWEEN_REQUEST_AND_DEPARTURE`

Default value for the minimum time between the request and the departure time.

## 9.3 AIRTSP::OnDParserHelper Namespace Reference

### Classes

- struct [ParserSemanticAction](#)

- struct [storeOrigin](#)
- struct [storeDestination](#)
- struct [storeDateRangeStart](#)
- struct [storeDateRangeEnd](#)
- struct [storeStartRangeTime](#)
- struct [storeEndRangeTime](#)
- struct [storeAirlineCode](#)
- struct [storeClassCode](#)
- struct [doEndOnD](#)
- struct [OnDParser](#)

## Functions

- [chset\\_t alpha\\_cap\\_set\\_p](#) ("A-Z")
- [repeat\\_p\\_t airport\\_p](#) ([chset\\_t](#)("0-9A-Z").[derived\(\)](#), 3, 3)
- [repeat\\_p\\_t airline\\_code\\_p](#) ([alpha\\_cap\\_set\\_p](#).[derived\(\)](#), 2, 3)
- [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)
- [bounded2\\_p\\_t hours\\_p](#) ([uint2\\_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 class\\_code\\_p](#) ("A-Z")

## Variables

- [uint2\\_p\\_t uint2\\_p](#)
- [uint4\\_p\\_t uint4\\_p](#)
- [uint1\\_4\\_p\\_t uint1\\_4\\_p](#)

### 9.3.1 Function Documentation

#### 9.3.1.1 [chset\\_t](#) AIRTSP::OnDParserHelper::alpha\_cap\_set\_p ("A-Z")

Sequence of (capital) alphabetic characters: [chset\\_p](#)("A-Z")

#### 9.3.1.2 [repeat\\_p\\_t](#) AIRTSP::OnDParserHelper::airport\_p ([chset\\_t](#)("0-9A-Z").[derived\(\)](#), 3, 3)

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

Referenced by AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::definition().

#### 9.3.1.3 [repeat\\_p\\_t](#) AIRTSP::OnDParserHelper::airline\_code\_p ([alpha\\_cap\\_set\\_p](#). *derived()*, 2, 3)

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

Referenced by AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::definition().

**9.3.1.4 [bounded4\\_p\\_t](#) AIRTSP::OnDParserHelper::year\_p (uint4\_p. *derived()*, 2000u, 2099u)**

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

Referenced by AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::definition().

**9.3.1.5 [bounded2\\_p\\_t](#) AIRTSP::OnDParserHelper::month\_p (uint2\_p. *derived()*, 1u, 12u)**

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

Referenced by AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::definition().

**9.3.1.6 [bounded2\\_p\\_t](#) AIRTSP::OnDParserHelper::day\_p (uint2\_p. *derived()*, 1u, 31u)**

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

Referenced by AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::definition().

**9.3.1.7 [bounded2\\_p\\_t](#) AIRTSP::OnDParserHelper::hours\_p (uint2\_p. *derived()*, 0u, 23u)**

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

Referenced by AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::definition().

**9.3.1.8 [bounded2\\_p\\_t](#) AIRTSP::OnDParserHelper::minutes\_p (uint2\_p. *derived()*, 0u, 59u)**

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

Referenced by AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::definition().

**9.3.1.9 [bounded2\\_p\\_t](#) AIRTSP::OnDParserHelper::seconds\_p (uint2\_p. *derived()*, 0u, 59u)**

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

Referenced by AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::definition().

**9.3.1.10 [chset\\_t](#) AIRTSP::OnDParserHelper::class\_code\_p ("A-Z")**

Class Code Parser: chset\_p("A-Z")

Referenced by AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::definition().

**9.3.2 Variable Documentation****9.3.2.1 [uint2\\_p\\_t](#) AIRTSP::OnDParserHelper::uint2\_p**

2-digit-integer parser

Definition at line 215 of file OnDParserHelper.cpp.

**9.3.2.2 [uint4\\_p\\_t](#) AIRTSP::OnDParserHelper::uint4\_p**

4-digit-integer parser

Definition at line 218 of file OnDParserHelper.cpp.

**9.3.2.3 uint1\_4\_p\_t AIRTSP::OnDParserHelper::uint1\_4\_p**

Up-to-4-digit-integer parser

Definition at line 221 of file OnDParserHelper.cpp.

**9.4 AIRTSP::ScheduleParserHelper Namespace Reference****Classes**

- struct [ParserSemanticAction](#)
- struct [storeAirlineCode](#)
- struct [storeFlightNumber](#)
- struct [storeDateRangeStart](#)
- struct [storeDateRangeEnd](#)
- struct [storeDow](#)
- struct [storeLegBoardingPoint](#)
- struct [storeLegOffPoint](#)
- struct [storeOperatingAirlineCode](#)
- struct [storeOperatingFlightNumber](#)
- struct [storeBoardingTime](#)
- struct [storeOffTime](#)
- struct [storeElapsedTime](#)
- struct [storeLegCabinCode](#)
- struct [storeCapacity](#)
- struct [storeSegmentSpecificity](#)
- struct [storeSegmentBoardingPoint](#)
- struct [storeSegmentOffPoint](#)
- struct [storeSegmentCabinCode](#)
- struct [storeClasses](#)
- struct [storeFamilyCode](#)
- struct [storeFRAT5CurveKey](#)
- struct [storeFFDisutilityCurveKey](#)
- struct [storeFClasses](#)
- struct [doEndFlight](#)
- struct [FlightPeriodParser](#)

**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)
- [bounded2\\_p\\_t hours\\_p](#) ([uint2\\_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")
- [repeat\\_p\\_t key\\_p](#) ([chset\\_t](#)("0-9A-Z").[derived](#)(), 1, 10)
- [repeat\\_p\\_t class\\_code\\_list\\_p](#) ([chset\\_t](#)("A-Z").[derived](#)(), 1, 26)



## Variables

- [int1\\_p\\_t](#) [int1\\_p](#)
- [uint2\\_p\\_t](#) [uint2\\_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](#) AIRTSP::ScheduleParserHelper::airline\_code\_p ([chset\\_t](#)("0-9A-Z").derived(), 2, 3)

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

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

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

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

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

#### 9.4.1.3 [bounded4\\_p\\_t](#) AIRTSP::ScheduleParserHelper::year\_p (uint4\_p. derived(), 2000u, 2099u)

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

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

#### 9.4.1.4 [bounded2\\_p\\_t](#) AIRTSP::ScheduleParserHelper::month\_p (uint2\_p. derived(), 1u, 12u)

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

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

#### 9.4.1.5 [bounded2\\_p\\_t](#) AIRTSP::ScheduleParserHelper::day\_p (uint2\_p. derived(), 1u, 31u)

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

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

#### 9.4.1.6 [repeat\\_p\\_t](#) AIRTSP::ScheduleParserHelper::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 AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**9.4.1.7 repeat\_p\_t AIRTSP::ScheduleParserHelper::airport\_p (chset\_t("0-9A-Z").derived(), 3, 3)**

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

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**9.4.1.8 bounded2\_p\_t AIRTSP::ScheduleParserHelper::hours\_p (uint2\_p. derived(), 0u, 23u)**

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

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**9.4.1.9 bounded2\_p\_t AIRTSP::ScheduleParserHelper::minutes\_p (uint2\_p. derived(), 0u, 59u)**

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

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**9.4.1.10 bounded2\_p\_t AIRTSP::ScheduleParserHelper::seconds\_p (uint2\_p. derived(), 0u, 59u)**

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

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**9.4.1.11 chset\_t AIRTSP::ScheduleParserHelper::cabin\_code\_p ("A-Z")**

Cabin Code Parser: chset\_p("A-Z")

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**9.4.1.12 repeat\_p\_t AIRTSP::ScheduleParserHelper::key\_p (chset\_t("0-9A-Z").derived(), 1, 10)**

Key Parser: repeat\_p(1,10)[chset\_p("0-9A-Z")]

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**9.4.1.13 repeat\_p\_t AIRTSP::ScheduleParserHelper::class\_code\_list\_p (chset\_t("A-Z").derived(), 1, 26)**

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

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**9.4.2 Variable Documentation****9.4.2.1 int1\_p\_t AIRTSP::ScheduleParserHelper::int1\_p**

1-digit-integer parser

Definition at line 473 of file ScheduleParserHelper.cpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

#### 9.4.2.2 [uint2\\_p\\_t AIRTSP::ScheduleParserHelper::uint2\\_p](#)

2-digit-integer parser

Definition at line 476 of file ScheduleParserHelper.cpp.

#### 9.4.2.3 [uint4\\_p\\_t AIRTSP::ScheduleParserHelper::uint4\\_p](#)

4-digit-integer parser

Definition at line 479 of file ScheduleParserHelper.cpp.

#### 9.4.2.4 [uint1\\_4\\_p\\_t AIRTSP::ScheduleParserHelper::uint1\\_4\\_p](#)

Up-to-4-digit-integer parser

Definition at line 482 of file ScheduleParserHelper.cpp.

#### 9.4.2.5 [int1\\_p\\_t AIRTSP::ScheduleParserHelper::family\\_code\\_p](#)

Family code parser

Definition at line 518 of file ScheduleParserHelper.cpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

## 9.5 boost Namespace Reference

Forward declarations.

### Namespaces

- namespace [serialization](#)

### 9.5.1 Detailed Description

Forward declarations.

## 9.6 boost::serialization Namespace Reference

## 9.7 boost::spirit::classic Namespace Reference

### 9.7.1 Detailed Description

Sample guadeloupe rio de janeiro 07/22/2009 +aa -ua 2 adults 1 dog

Grammar: search\_string ::= places [dates] (preferred\_airlines) (passengers) dates ::= board\_date [off\_date]  
 places ::= [board\_place] off\_place board\_place ::= place\_elements off\_place ::= place\_elements place\_  
 elements ::= country | city | airport country ::= country\_code | country\_name city ::= city\_code | city\_name  
 airport ::= airport\_code | airport\_name preferred\_airlines ::= [+|-] airline\_code | airline\_name passen-  
 gers ::= adult\_number adult\_description [child\_number child\_description] [pet\_number pet\_description]  
 adult\_description ::= 'adult' | 'adults' | 'pax' | 'passengers' child\_description ::= 'child' | 'children' | 'kid'  
 | 'kids' pet\_description ::= 'dog' | 'dogs' | 'cat' | 'cats'

## 9.8 stdair Namespace Reference

Forward declarations.

### 9.8.1 Detailed Description

Forward declarations.

## 10 AirTSP Class Documentation

### 10.1 airtsp::Airline\_T Struct Reference

```
#include <airtsp/batches/BookingRequestParser.hpp>
```

#### Public Member Functions

- [Airline\\_T\(\)](#)
- void [display\(\)](#) const

#### Public Attributes

- bool [\\_isPreferred](#)
- std::string [\\_name](#)
- std::string [\\_code](#)

#### 10.1.1 Detailed Description

Airline.

Definition at line 52 of file BookingRequestParser.hpp.

#### 10.1.2 Constructor & Destructor Documentation

##### 10.1.2.1 airtsp::Airline\_T::Airline\_T() [inline]

Constructor.

Definition at line 58 of file BookingRequestParser.hpp.

### 10.1.3 Member Function Documentation

#### 10.1.3.1 void airtsp::Airline\_T::display () const [inline]

Definition at line 60 of file BookingRequestParser.hpp.

References `_code`, `_isPreferred`, and `_name`.

### 10.1.4 Member Data Documentation

#### 10.1.4.1 bool airtsp::Airline\_T::\_isPreferred

Definition at line 54 of file BookingRequestParser.hpp.

Referenced by `display()`, and `airtsp::store_airline_sign::operator()`.

#### 10.1.4.2 std::string airtsp::Airline\_T::\_name

Definition at line 55 of file BookingRequestParser.hpp.

Referenced by `display()`, and `airtsp::store_airline_name::operator()`.

#### 10.1.4.3 std::string airtsp::Airline\_T::\_code

Definition at line 56 of file BookingRequestParser.hpp.

Referenced by `display()`, and `airtsp::store_airline_code::operator()`.

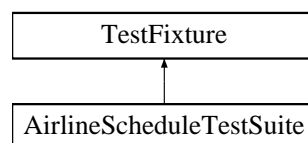
The documentation for this struct was generated from the following file:

- [airtsp/batches/BookingRequestParser.hpp](#)

## 10.2 AirlineScheduleTestSuite Class Reference

```
#include <test/airtsp/AirlineScheduleTestSuite.hpp>
```

Inheritance diagram for AirlineScheduleTestSuite::



### Public Member Functions

- void [externalMemoryManagement](#) ()
- void [scheduleParsing](#) ()
- [AirlineScheduleTestSuite](#) ()

### Protected Attributes

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

### 10.2.1 Detailed Description

Definition at line 6 of file AirlineScheduleTestSuite.hpp.

### 10.2.2 Constructor & Destructor Documentation

#### 10.2.2.1 AirlineScheduleTestSuite::AirlineScheduleTestSuite ()

Constructor.

### 10.2.3 Member Function Documentation

#### 10.2.3.1 void AirlineScheduleTestSuite::externalMemoryManagement ()

Test the Optimisation functionality.

The code is aimed at testing the initialization of airline inventory-related objects which are mainly implemented in the stdair library. That means the memory allocation of these objects will be managed by the calling project and not by the called project.

#### 10.2.3.2 void AirlineScheduleTestSuite::scheduleParsing ()

### 10.2.4 Member Data Documentation

#### 10.2.4.1 std::stringstream AirlineScheduleTestSuite::\_describeKey [protected]

Definition at line 26 of file AirlineScheduleTestSuite.hpp.

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

- test/airtsp/[AirlineScheduleTestSuite.hpp](#)

## 10.3 AIRTSP::AIRTSP\_Service Class Reference

Interface for the Airtsp Services.

```
#include <airtsp/AIRTSP_Service.hpp>
```

### Public Member Functions

- [AIRTSP\\_Service](#) (const stdair::BasLogParams &, const stdair::BasDBParams &)
- [AIRTSP\\_Service](#) (const stdair::BasLogParams &)
- [AIRTSP\\_Service](#) (stdair::STDAIR\_ServicePtr\_T ioSTDAIR\_ServicePtr)
- void [parseAndLoad](#) (const stdair::ScheduleFilePath &)
- void [parseAndLoad](#) (const stdair::ScheduleFilePath &, const stdair::ODFilePath &)
- [~AIRTSP\\_Service](#) ()
- void [buildSampleBom](#) ()
- void [clonePersistentBom](#) ()
- void [buildComplementaryLinks](#) (stdair::BomRoot &)
- void [buildSegmentPathList](#) (stdair::TravelSolutionList\_T &, const stdair::BookingRequestStruct &)
- void [simulate](#) ()

- std::string [jsonExportFlightDateObjects](#) (const stdair::AirlineCode\_T &, const stdair::FlightNumber\_T &, const stdair::Date\_T & iDepartureDate) const
- std::string [csvDisplay](#) () const
- std::string [csvDisplay](#) (const stdair::AirlineCode\_T &, const stdair::FlightNumber\_T &, const stdair::Date\_T & iDepartureDate) const

### 10.3.1 Detailed Description

Interface for the Airtsp Services.

Definition at line 32 of file AIRTSP\_Service.hpp.

### 10.3.2 Constructor & Destructor Documentation

#### 10.3.2.1 AIRTSP::AIRTSP\_Service::AIRTSP\_Service (const stdair::BasLogParams &, const stdair::BasDBParams &)

Constructor.

The initAirtspService() 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.

Definition at line 62 of file AIRTSP\_Service.cpp.

#### 10.3.2.2 AIRTSP::AIRTSP\_Service::AIRTSP\_Service (const stdair::BasLogParams &)

Constructor.

The initAirtspService() 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.

Definition at line 42 of file AIRTSP\_Service.cpp.

#### 10.3.2.3 AIRTSP::AIRTSP\_Service::AIRTSP\_Service (stdair::STDAIR\_ServicePtr\_T io-STDAIR\_ServicePtr)

Constructor.

The initAirtspService() method is called; see the corresponding documentation for more details.

Moreover, as no reference on any output stream 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 [AIRTSP\\_Service](#) is itself being initialised by another library service such as SIMCRS\_Service).

**Parameters:**

*stdair::STDAIR\_ServicePtr\_T* Reference on the STDAIR service.

Definition at line 84 of file AIRTSP\_Service.cpp.

**10.3.2.4 AIRTSP::AIRTSP\_Service::~~AIRTSP\_Service ()**

Destructor.

Definition at line 100 of file AIRTSP\_Service.cpp.

**10.3.3 Member Function Documentation****10.3.3.1 void AIRTSP::AIRTSP\_Service::parseAndLoad (const stdair::ScheduleFilePath &)**

Parse the schedule input file and load it into memory.

The CSV file, describing the airline schedule for the simulator, is parsed and instantiated in memory accordingly.

**Parameters:**

*const* stdair::ScheduleFilePath& Filename of the input schedule file.

Definition at line 178 of file AIRTSP\_Service.cpp.

References buildComplementaryLinks(), clonePersistentBom(), AIRTSP::ScheduleParser::generateInventories(), AIRTSP::AIRTSP\_ServiceContext::getOwnStdairServiceFlag(), and AIRTSP::AIRTSP\_ServiceContext::getSTDAIR\_Service().

Referenced by main(), and parseAndLoad().

**10.3.3.2 void AIRTSP::AIRTSP\_Service::parseAndLoad (const stdair::ScheduleFilePath &, const stdair::ODFilePath &)**

Parse the schedule and O&D input files, and load them into memory.

The CSV files, describing the airline schedule and the O&Ds for the simulator, are parsed and instantiated in memory accordingly.

**Parameters:**

*const* stdair::ScheduleFilePath& Filename of the input schedule file.

*const* stdair::ODFilePath& Filename of the input O&D file.

Definition at line 230 of file AIRTSP\_Service.cpp.

References buildComplementaryLinks(), AIRTSP::OnDParser::generateOnDPeriods(), AIRTSP::AIRTSP\_ServiceContext::getOwnStdairServiceFlag(), AIRTSP::AIRTSP\_ServiceContext::getSTDAIR\_Service(), and parseAndLoad().

**10.3.3.3 void AIRTSP::AIRTSP\_Service::buildSampleBom ()**

Build a sample BOM tree, and attach it to the BomRoot instance.

The BOM tree is based on two actual inventories (one for BA, another for AF). Each inventory contains one flight. One of those flights has two legs (and therefore three segments).



Definition at line 287 of file AIRTSP\_Service.cpp.

References buildComplementaryLinks(), clonePersistentBom(), AIRTSP::AIRTSP\_ServiceContext::getOwnStdairServiceFlag(), and AIRTSP::AIRTSP\_ServiceContext::getSTDAIR\_Service().

Referenced by main().

#### 10.3.3.4 void AIRTSP::AIRTSP\_Service::clonePersistentBom ()

Clone the persistent BOM object.

Definition at line 344 of file AIRTSP\_Service.cpp.

References buildComplementaryLinks(), AIRTSP::AIRTSP\_ServiceContext::getOwnStdairServiceFlag(), and AIRTSP::AIRTSP\_ServiceContext::getSTDAIR\_Service().

Referenced by buildSampleBom(), and parseAndLoad().

#### 10.3.3.5 void AIRTSP::AIRTSP\_Service::buildComplementaryLinks (stdair::BomRoot &)

Build all the complementary links in the given bom root object.

Definition at line 384 of file AIRTSP\_Service.cpp.

References AIRTSP::SegmentPathGenerator::createSegmentPathNetwork().

Referenced by buildSampleBom(), clonePersistentBom(), and parseAndLoad().

#### 10.3.3.6 void AIRTSP::AIRTSP\_Service::buildSegmentPathList (stdair::TravelSolutionList\_T &, const stdair::BookingRequestStruct &)

Calculate and return a list of travel solutions corresponding to a given product demand.

Definition at line 498 of file AIRTSP\_Service.cpp.

References AIRTSP::AIRTSP\_ServiceContext::display(), and AIRTSP::AIRTSP\_ServiceContext::getSTDAIR\_Service().

Referenced by main().

#### 10.3.3.7 void AIRTSP::AIRTSP\_Service::simulate ()

Perform a small simulation, which uses the Customer Choice Model (CCM).

Currently, that method does nothing.

Definition at line 470 of file AIRTSP\_Service.cpp.

References AIRTSP::AIRTSP\_ServiceContext::display(), AIRTSP::AIRTSP\_ServiceContext::getSTDAIR\_Service(), and AIRTSP::Simulator::simulate().

#### 10.3.3.8 std::string AIRTSP::AIRTSP\_Service::jsonExportFlightDateObjects (const stdair::AirlineCode\_T &, const stdair::FlightNumber\_T &, const stdair::Date\_T & iDepartureDate) const

Recursively dump, in the returned string and in JSON format, the flight-period corresponding to the parameters given as input.

##### Parameters:

**const** stdair::AirlineCode\_T& Airline code of the flight to dump.

**const** stdair::FlightNumber\_T& Flight number of the flight to dump.

**const** stdair::Date\_T& Departure date of a flight within the flight period to dump.

#### Returns:

std::string Output string in which the BOM tree is JSON-ified.

Definition at line 401 of file AIRTSP\_Service.cpp.

References AIRTSP::AIRTSP\_ServiceContext::getSTDAIR\_Service().

#### 10.3.3.9 std::string AIRTSP::AIRTSP\_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 425 of file AIRTSP\_Service.cpp.

References AIRTSP::AIRTSP\_ServiceContext::getSTDAIR\_Service().

#### 10.3.3.10 std::string AIRTSP::AIRTSP\_Service::csvDisplay (const stdair::AirlineCode\_T &, const stdair::FlightNumber\_T &, const stdair::Date\_T & iDepartureDate) const

Recursively display (dump in the returned string) the flight-period corresponding to the parameters given as input.

#### Parameters:

**const** stdair::AirlineCode\_T& Airline code of the flight period to display.

**const** stdair::FlightNumber\_T& Flight number of the flight to display.

**const** stdair::Date\_T& Departure date of a flight within the flight-period to display.

#### Returns:

std::string Output string in which the BOM tree is logged/dumped.

Definition at line 447 of file AIRTSP\_Service.cpp.

References AIRTSP::AIRTSP\_ServiceContext::getSTDAIR\_Service().

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

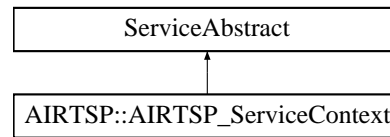
- [airtsp/AIRTSP\\_Service.hpp](#)
- [airtsp/service/AIRTSP\\_Service.cpp](#)

## 10.4 AIRTSP::AIRTSP\_ServiceContext Class Reference

Class holding the context of the Airtsp services.

```
#include <airtsp/service/AIRTSP_ServiceContext.hpp>
```

Inheritance diagram for AIRTSP::AIRTSP\_ServiceContext::



### Friends

- class [AIRTSP\\_Service](#)
- class [FacAIRTSPServiceContext](#)

#### 10.4.1 Detailed Description

Class holding the context of the Airtsp services.

Definition at line 22 of file AIRTSP\_ServiceContext.hpp.

#### 10.4.2 Friends And Related Function Documentation

##### 10.4.2.1 friend class [AIRTSP\\_Service](#) [friend]

The [AIRTSP\\_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 28 of file AIRTSP\_ServiceContext.hpp.

##### 10.4.2.2 friend class [FacAIRTSPServiceContext](#) [friend]

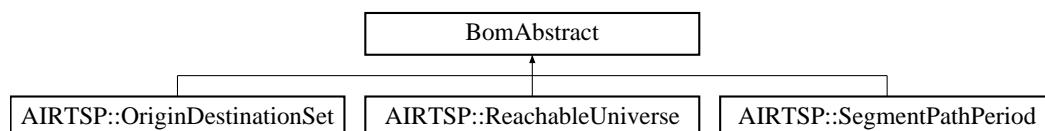
Definition at line 29 of file AIRTSP\_ServiceContext.hpp.

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

- airtsp/service/[AIRTSP\\_ServiceContext.hpp](#)
- airtsp/service/[AIRTSP\\_ServiceContext.cpp](#)

## 10.5 BomAbstract Class Reference

Inheritance diagram for BomAbstract::



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

- airtsp/bom/[SegmentPathPeriod.hpp](#)
- airtsp/bom/[OriginDestinationSet.hpp](#)
- airtsp/bom/[ReachableUniverse.hpp](#)

## 10.6 AIRTSP::BomDisplay Class Reference

Utility class to display AirTSP objects with a pretty format.

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

### Static Public Member Functions

- static std::string [csvDisplay](#) (const stdair::BomRoot &)
- static void [csvDisplay](#) (std::ostream &, const [ReachableUniverse](#) &)

### 10.6.1 Detailed Description

Utility class to display AirTSP objects with a pretty format.

Definition at line 26 of file BomDisplay.hpp.

### 10.6.2 Member Function Documentation

#### 10.6.2.1 std::string AIRTSP::BomDisplay::csvDisplay (const stdair::BomRoot &) [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* stdair::EventQueue& Root of the BOM tree to be displayed.

Definition at line 43 of file BomDisplay.cpp.

#### 10.6.2.2 void AIRTSP::BomDisplay::csvDisplay (std::ostream &, const [ReachableUniverse](#) &) [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* [ReachableUniverse](#)& Root of the BOM tree to be displayed.

Definition at line 81 of file BomDisplay.cpp.

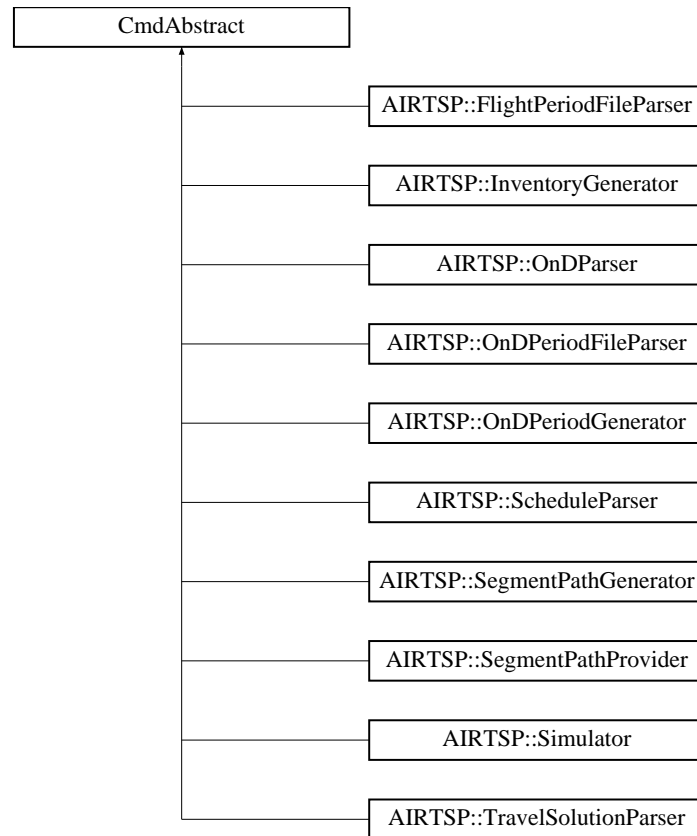
References [AIRTSP::ReachableUniverse::toString\(\)](#).

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

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

## 10.7 CmdAbstract Class Reference

Inheritance diagram for CmdAbstract::

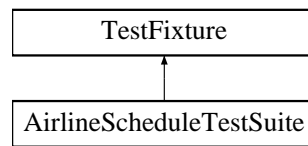


The documentation for this class was generated from the following files:

- [airtsp/command/Simulator.hpp](#)
- [airtsp/command/ScheduleParserHelper.hpp](#)
- [airtsp/command/SegmentPathProvider.hpp](#)
- [airtsp/command/OnDParserHelper.hpp](#)
- [airtsp/command/SegmentPathGenerator.hpp](#)
- [airtsp/command/InventoryGenerator.hpp](#)
- [airtsp/command/OnDPeriodGenerator.hpp](#)
- [airtsp/command/TravelSolutionParser.hpp](#)
- [airtsp/command/ScheduleParser.hpp](#)
- [airtsp/command/OnDParser.hpp](#)

## 10.8 TestFixture Class Reference

Inheritance diagram for TestFixture::



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

- test/airtsp/[AirlineScheduleTestSuite.hpp](#)

## 10.9 airtsp::Date\_T Struct Reference

```
#include <airtsp/batches/BookingRequestParser.hpp>
```

### Public Member Functions

- [Date\\_T](#) ()
- void [display](#) () const
- boost::gregorian::date [getDate](#) () const

### Public Attributes

- boost::gregorian::date [\\_date](#)
- unsigned int [\\_reldays](#)
- unsigned int [\\_day](#)
- unsigned int [\\_month](#)
- unsigned int [\\_year](#)

### 10.9.1 Detailed Description

Date.

Definition at line 27 of file BookingRequestParser.hpp.

### 10.9.2 Constructor & Destructor Documentation

#### 10.9.2.1 airtsp::Date\_T::Date\_T () [inline]

Constructor.

Definition at line 35 of file BookingRequestParser.hpp.

### 10.9.3 Member Function Documentation

#### 10.9.3.1 void airtsp::Date\_T::display () const [inline]

Definition at line 37 of file BookingRequestParser.hpp.

References [\\_date](#), [\\_day](#), [\\_month](#), [\\_reldays](#), and [\\_year](#).

**10.9.3.2 boost::gregorian::date airtsp::Date\_T::getDate () const [inline]**

Set the date from the staging details.

Definition at line 43 of file BookingRequestParser.hpp.

References `_day`, `_month`, and `_year`.

Referenced by `airtsp::store_date::operator()`.

**10.9.4 Member Data Documentation****10.9.4.1 boost::gregorian::date airtsp::Date\_T::\_date**

Definition at line 29 of file BookingRequestParser.hpp.

Referenced by `display()`, and `airtsp::store_date::operator()`.

**10.9.4.2 unsigned int airtsp::Date\_T::\_reldays**

Definition at line 30 of file BookingRequestParser.hpp.

Referenced by `display()`.

**10.9.4.3 unsigned int airtsp::Date\_T::\_day**

Definition at line 31 of file BookingRequestParser.hpp.

Referenced by `display()`, and `getDate()`.

**10.9.4.4 unsigned int airtsp::Date\_T::\_month**

Definition at line 32 of file BookingRequestParser.hpp.

Referenced by `display()`, and `getDate()`.

**10.9.4.5 unsigned int airtsp::Date\_T::\_year**

Definition at line 33 of file BookingRequestParser.hpp.

Referenced by `display()`, and `getDate()`.

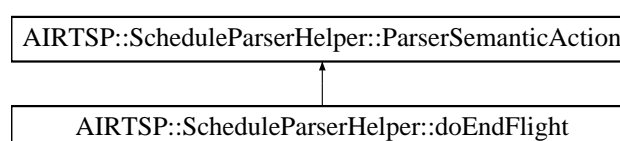
The documentation for this struct was generated from the following file:

- `airtsp/batches/BookingRequestParser.hpp`

**10.10 AIRTSP::ScheduleParserHelper::doEndFlight Struct Reference**

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::doEndFlight::



### Public Member Functions

- [doEndFlight](#) (stdair::BomRoot &, [FlightPeriodStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- stdair::BomRoot & [\\_bomRoot](#)
- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.10.1 Detailed Description

Mark the end of the flight-period parsing.

Definition at line 224 of file ScheduleParserHelper.hpp.

#### 10.10.2 Constructor & Destructor Documentation

##### 10.10.2.1 AIRTSP::ScheduleParserHelper::doEndFlight::doEndFlight (stdair::BomRoot &, [FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 440 of file ScheduleParserHelper.cpp.

#### 10.10.3 Member Function Documentation

##### 10.10.3.1 void AIRTSP::ScheduleParserHelper::doEndFlight::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 448 of file ScheduleParserHelper.cpp.

References [\\_bomRoot](#), [AIRTSP::LegStruct::\\_cabinList](#), [AIRTSP::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#), [AIRTSP::FlightPeriodStruct::\\_itLeg](#), [AIRTSP::FlightPeriodStruct::\\_legAlreadyDefined](#), [AIRTSP::FlightPeriodStruct::\\_legList](#), and [AIRTSP::FlightPeriodStruct::describe\(\)](#).

#### 10.10.4 Member Data Documentation

##### 10.10.4.1 stdair::BomRoot& [AIRTSP::ScheduleParserHelper::doEndFlight::\\_bomRoot](#)

Actor Specific Context.

Definition at line 230 of file ScheduleParserHelper.hpp.

Referenced by [operator\(\)](#).

##### 10.10.4.2 [FlightPeriodStruct](#)& [AIRTSP::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#) [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.



Referenced by operator(), AIRTSP::ScheduleParserHelper::storeFCClasses::operator(), AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

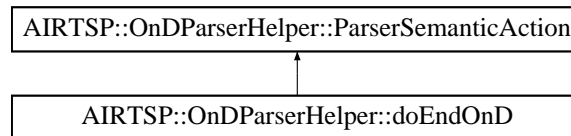
The documentation for this struct was generated from the following files:

- airtsp/command/[ScheduleParserHelper.hpp](#)
- airtsp/command/[ScheduleParserHelper.cpp](#)

## 10.11 AIRTSP::OnDParserHelper::doEndOnD Struct Reference

```
#include <airtsp/command/OnDParserHelper.hpp>
```

Inheritance diagram for AIRTSP::OnDParserHelper::doEndOnD:



### Public Member Functions

- [doEndOnD](#) (stdair::BomRoot &, [OnDPeriodStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- stdair::BomRoot & [\\_bomRoot](#)
- [OnDPeriodStruct](#) & [\\_onDPeriod](#)

#### 10.11.1 Detailed Description

Mark the end of the O&D parsing.

Definition at line 106 of file OnDParserHelper.hpp.

### 10.11.2 Constructor & Destructor Documentation

#### 10.11.2.1 AIRTSP::OnDParserHelper::doEndOnD::doEndOnD (stdair::BomRoot &, [OnDPeriodStruct](#) &)

Actor Constructor.

Definition at line 193 of file OnDParserHelper.cpp.

### 10.11.3 Member Function Documentation

#### 10.11.3.1 void AIRTSP::OnDParserHelper::doEndOnD::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStr-End) const

Actor Function (functor).

Definition at line 199 of file OnDParserHelper.cpp.

References `_bomRoot`, and `AIRTSP::OnDParserHelper::ParserSemanticAction::_onDPeriod`.

### 10.11.4 Member Data Documentation

#### 10.11.4.1 stdair::BomRoot& [AIRTSP::OnDParserHelper::doEndOnD::\\_bomRoot](#)

Actor Specific Context.

Definition at line 112 of file OnDParserHelper.hpp.

Referenced by `operator()`.

#### 10.11.4.2 [OnDPeriodStruct](#)& [AIRTSP::OnDParserHelper::ParserSemanticAction::\\_onDPeriod](#) [inherited]

Actor Context.

Definition at line 38 of file OnDParserHelper.hpp.

Referenced by `operator()`, `AIRTSP::OnDParserHelper::storeClassCode::operator()`, `AIRTSP::OnDParserHelper::storeAirlineCode::operator()`, `AIRTSP::OnDParserHelper::storeEndRangeTime::operator()`, `AIRTSP::OnDParserHelper::storeStartRangeTime::operator()`, `AIRTSP::OnDParserHelper::storeDateRangeEnd::operator()`, `AIRTSP::OnDParserHelper::storeDateRangeStart::operator()`, `AIRTSP::OnDParserHelper::storeDestination::operator()`, and `AIRTSP::OnDParserHelper::storeOrigin::operator()`.

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

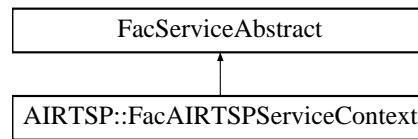
- `airtsp/command/OnDParserHelper.hpp`
- `airtsp/command/OnDParserHelper.cpp`

## 10.12 AIRTSP::FacAIRTSPServiceContext Class Reference

Factory for the service context.

```
#include <airtsp/factory/FacAIRTSPServiceContext.hpp>
```

Inheritance diagram for AIRTSP::FacAIRTSPServiceContext:



### Public Member Functions

- [~FacAIRTSPServiceContext \(\)](#)
- [AIRTSP\\_ServiceContext & create \(\)](#)

### Static Public Member Functions

- static [FacAIRTSPServiceContext & instance \(\)](#)

### Protected Member Functions

- [FacAIRTSPServiceContext \(\)](#)

#### 10.12.1 Detailed Description

Factory for the service context.

Definition at line 19 of file FacAIRTSPServiceContext.hpp.

#### 10.12.2 Constructor & Destructor Documentation

##### 10.12.2.1 AIRTSP::FacAIRTSPServiceContext::~~FacAIRTSPServiceContext ()

Destructor.

The Destruction put the `_instance` to NULL in order to be clean for the next [FacAIRTSPServiceContext::instance\(\)](#).

Definition at line 17 of file FacAIRTSPServiceContext.cpp.

##### 10.12.2.2 AIRTSP::FacAIRTSPServiceContext::FacAIRTSPServiceContext () [inline, protected]

Default Constructor.

This constructor is protected in order to ensure the singleton pattern.

Definition at line 54 of file FacAIRTSPServiceContext.hpp.

Referenced by `instance()`.

#### 10.12.3 Member Function Documentation

##### 10.12.3.1 [FacAIRTSPServiceContext](#) & AIRTSP::FacAIRTSPServiceContext::instance () [static]

Provide the unique instance.

The singleton is instantiated when first used.

#### Returns:

FacServiceContext&

Definition at line 22 of file FacAIRTSPServiceContext.cpp.

References FacAIRTSPServiceContext().

#### 10.12.3.2 [AIRTSP\\_ServiceContext](#) & AIRTSP::FacAIRTSPServiceContext::create ()

Create a new ServiceContext object.

This new object is added to the list of instantiated objects.

#### Returns:

ServiceContext& The newly created object.

Definition at line 34 of file FacAIRTSPServiceContext.cpp.

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

- [airtsp/factory/FacAIRTSPServiceContext.hpp](#)
- [airtsp/factory/FacAIRTSPServiceContext.cpp](#)

## 10.13 AIRTSP::FacServiceAbstract Class Reference

```
#include <airtsp/factory/FacServiceAbstract.hpp>
```

### Public Types

- typedef std::vector< [ServiceAbstract](#) \* > [ServicePool\\_T](#)

### Public Member Functions

- virtual [~FacServiceAbstract](#) ()
- void [clean](#) ()

### Protected Member Functions

- [FacServiceAbstract](#) ()

### Protected Attributes

- [ServicePool\\_T \\_pool](#)

#### 10.13.1 Detailed Description

Base class for the (Service) Factory layer.

Definition at line 16 of file FacServiceAbstract.hpp.

### 10.13.2 Member Typedef Documentation

#### 10.13.2.1 `typedef std::vector<ServiceAbstract*> AIRTSP::FacServiceAbstract::ServicePool_T`

Define the list (pool) of Service objects.

Definition at line 20 of file FacServiceAbstract.hpp.

### 10.13.3 Constructor & Destructor Documentation

#### 10.13.3.1 `AIRTSP::FacServiceAbstract::~~FacServiceAbstract ()` `[virtual]`

Destructor.

Definition at line 13 of file FacServiceAbstract.cpp.

#### 10.13.3.2 `AIRTSP::FacServiceAbstract::FacServiceAbstract ()` `[inline, protected]`

Default Constructor.

This constructor is protected to ensure the class is abstract.

Definition at line 31 of file FacServiceAbstract.hpp.

### 10.13.4 Member Function Documentation

#### 10.13.4.1 `void AIRTSP::FacServiceAbstract::clean ()`

Destroyed all the object instantiated by this factory.

Definition at line 18 of file FacServiceAbstract.cpp.

### 10.13.5 Member Data Documentation

#### 10.13.5.1 `ServicePool_T AIRTSP::FacServiceAbstract::_pool` `[protected]`

List of instantiated Business Objects

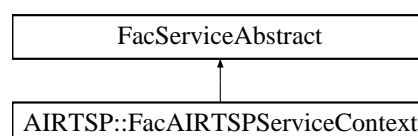
Definition at line 34 of file FacServiceAbstract.hpp.

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

- [airtsp/factory/FacServiceAbstract.hpp](#)
- [airtsp/factory/FacServiceAbstract.cpp](#)

## 10.14 FacServiceAbstract Class Reference

Inheritance diagram for FacServiceAbstract::



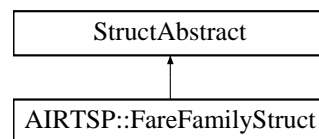
The documentation for this class was generated from the following file:

- airtsp/factory/[FacAIRTSPServiceContext.hpp](#)

## 10.15 AIRTSP::FareFamilyStruct Struct Reference

```
#include <airtsp/bom/FareFamilyStruct.hpp>
```

Inheritance diagram for AIRTSP::FareFamilyStruct::



### Public Member Functions

- [FareFamilyStruct](#) (const stdair::FamilyCode\_T &, const stdair::CurveKey\_T &, const stdair::CurveKey\_T &, const stdair::ClassList\_String\_T &)
- const std::string [describe](#) () const

### Public Attributes

- stdair::FamilyCode\_T [\\_familyCode](#)
- stdair::CurveKey\_T [\\_frat5CurveKey](#)
- stdair::CurveKey\_T [\\_ffDisutilityCurveKey](#)
- stdair::ClassList\_String\_T [\\_classes](#)

### 10.15.1 Detailed Description

Utility Structure for the parsing of fare family details.

Definition at line 17 of file FareFamilyStruct.hpp.

### 10.15.2 Constructor & Destructor Documentation

**10.15.2.1 AIRTSP::FareFamilyStruct::FareFamilyStruct (const stdair::FamilyCode\_T &, const stdair::CurveKey\_T &, const stdair::CurveKey\_T &, const stdair::ClassList\_String\_T &)**

Constructors.

Definition at line 14 of file FareFamilyStruct.cpp.

### 10.15.3 Member Function Documentation

**10.15.3.1 const std::string AIRTSP::FareFamilyStruct::describe () const**

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

Definition at line 23 of file FareFamilyStruct.cpp.

References [\\_classes](#), [\\_familyCode](#), [\\_ffDisutilityCurveKey](#), and [\\_frat5CurveKey](#).

### 10.15.4 Member Data Documentation

#### 10.15.4.1 stdair::FamilyCode\_T AIRTSP::FareFamilyStruct::\_familyCode

Definition at line 19 of file FareFamilyStruct.hpp.

Referenced by describe().

#### 10.15.4.2 stdair::CurveKey\_T AIRTSP::FareFamilyStruct::\_frat5CurveKey

Definition at line 20 of file FareFamilyStruct.hpp.

Referenced by describe().

#### 10.15.4.3 stdair::CurveKey\_T AIRTSP::FareFamilyStruct::\_ffDisutilityCurveKey

Definition at line 21 of file FareFamilyStruct.hpp.

Referenced by describe().

#### 10.15.4.4 stdair::ClassList\_String\_T AIRTSP::FareFamilyStruct::\_classes

Definition at line 22 of file FareFamilyStruct.hpp.

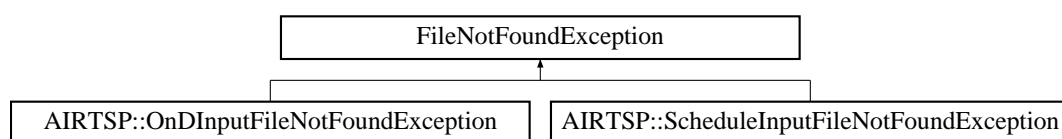
Referenced by describe().

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

- airtsp/bom/[FareFamilyStruct.hpp](#)
- airtsp/bom/[FareFamilyStruct.cpp](#)

## 10.16 FileNotFoundException Class Reference

Inheritance diagram for FileNotFoundException::



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

- airtsp/[AIRTSP\\_Types.hpp](#)

## 10.17 AIRTSP::FlagSaver Struct Reference

### Public Member Functions

- [FlagSaver](#) (std::ostream &oStream)
- [~FlagSaver](#) ()

### 10.17.1 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 22 of file BomDisplay.cpp.

### 10.17.2 Constructor & Destructor Documentation

#### 10.17.2.1 AIRTSP::FlagSaver::FlagSaver (std::ostream & oStream) [inline]

Constructor.

Definition at line 25 of file BomDisplay.cpp.

#### 10.17.2.2 AIRTSP::FlagSaver::~~FlagSaver () [inline]

Destructor.

Definition at line 30 of file BomDisplay.cpp.

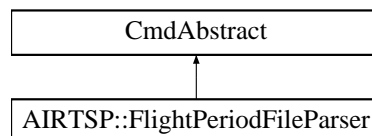
The documentation for this struct was generated from the following file:

- [airtsp/bom/BomDisplay.cpp](#)

## 10.18 AIRTSP::FlightPeriodFileParser Class Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::FlightPeriodFileParser::



### Public Member Functions

- [FlightPeriodFileParser](#) (stdair::BomRoot &ioBomRoot, const stdair::Filename\_T &iFilename)
- bool [generateInventories](#) ()

### 10.18.1 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 323 of file ScheduleParserHelper.hpp.



## 10.18.2 Constructor & Destructor Documentation

### 10.18.2.1 AIRTSP::FlightPeriodFileParser::FlightPeriodFileParser (stdair::BomRoot & ioBomRoot, const stdair::Filename\_T & iFilename)

Constructor.

Definition at line 716 of file ScheduleParserHelper.cpp.

## 10.18.3 Member Function Documentation

### 10.18.3.1 bool AIRTSP::FlightPeriodFileParser::generateInventories ()

Parse the input file and generate the Inventories.

Definition at line 753 of file ScheduleParserHelper.cpp.

Referenced by AIRTSP::ScheduleParser::generateInventories().

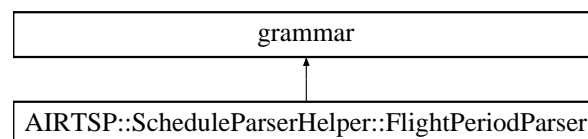
The documentation for this class was generated from the following files:

- airtsp/command/[ScheduleParserHelper.hpp](#)
- airtsp/command/[ScheduleParserHelper.cpp](#)

## 10.19 AIRTSP::ScheduleParserHelper::FlightPeriodParser Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::FlightPeriodParser::



## Public Member Functions

- [FlightPeriodParser](#) (stdair::BomRoot &, [FlightPeriodStruct](#) &)

## Public Attributes

- stdair::BomRoot & [\\_bomRoot](#)
- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

## Classes

- struct [definition](#)

## 10.19.1 Detailed Description

Grammar for the Flight-Period parser.

Definition at line 281 of file ScheduleParserHelper.hpp.

## 10.19.2 Constructor & Destructor Documentation

### 10.19.2.1 AIRTSP::ScheduleParserHelper::FlightPeriodParser::FlightPeriodParser (stdair::BomRoot &, [FlightPeriodStruct](#) &)

Definition at line 533 of file ScheduleParserHelper.cpp.

## 10.19.3 Member Data Documentation

### 10.19.3.1 stdair::BomRoot& [AIRTSP::ScheduleParserHelper::FlightPeriodParser::\\_bomRoot](#)

Definition at line 304 of file ScheduleParserHelper.hpp.

Referenced by `AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition()`.

### 10.19.3.2 [FlightPeriodStruct&](#) [AIRTSP::ScheduleParserHelper::FlightPeriodParser::\\_flight-Period](#)

Definition at line 305 of file ScheduleParserHelper.hpp.

Referenced by `AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition()`.

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

- airtsp/command/[ScheduleParserHelper.hpp](#)
- airtsp/command/[ScheduleParserHelper.cpp](#)

## 10.20 AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT > Struct Template Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

### Public Member Functions

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

### Public Attributes

- `boost::spirit::classic::rule< ScannerT > flight\_period\_list`
- `boost::spirit::classic::rule< ScannerT > flight\_period`
- `boost::spirit::classic::rule< ScannerT > not\_to\_be\_parsed`
- `boost::spirit::classic::rule< ScannerT > flight\_period\_end`
- `boost::spirit::classic::rule< ScannerT > flight\_key`
- `boost::spirit::classic::rule< ScannerT > airline\_code`
- `boost::spirit::classic::rule< ScannerT > flight\_number`
- `boost::spirit::classic::rule< ScannerT > date`
- `boost::spirit::classic::rule< ScannerT > dow`
- `boost::spirit::classic::rule< ScannerT > time`
- `boost::spirit::classic::rule< ScannerT > date\_offset`
- `boost::spirit::classic::rule< ScannerT > leg`

- [boost::spirit::classic::rule< ScannerT > leg\\_key](#)
- [boost::spirit::classic::rule< ScannerT > operating\\_leg\\_details](#)
- [boost::spirit::classic::rule< ScannerT > leg\\_details](#)
- [boost::spirit::classic::rule< ScannerT > leg\\_cabin\\_details](#)
- [boost::spirit::classic::rule< ScannerT > segment\\_section](#)
- [boost::spirit::classic::rule< ScannerT > segment\\_key](#)
- [boost::spirit::classic::rule< ScannerT > full\\_segment\\_cabin\\_details](#)
- [boost::spirit::classic::rule< ScannerT > segment\\_cabin\\_details](#)
- [boost::spirit::classic::rule< ScannerT > full\\_family\\_cabin\\_details](#)
- [boost::spirit::classic::rule< ScannerT > family\\_cabin\\_details](#)
- [boost::spirit::classic::rule< ScannerT > generic\\_segment](#)
- [boost::spirit::classic::rule< ScannerT > specific\\_segment\\_list](#)

### 10.20.1 Detailed Description

**template<typename ScannerT> struct AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >**

Definition at line 287 of file ScheduleParserHelper.hpp.

### 10.20.2 Constructor & Destructor Documentation

**10.20.2.1 template<typename ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition (FlightPeriodParser const & self)**

Definition at line 542 of file ScheduleParserHelper.cpp.

References AIRTSP::ScheduleParserHelper::FlightPeriodParser::\_bomRoot, AIRTSP::ScheduleParserHelper::FlightPeriodParser::\_flightPeriod, AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::airline\_code, AIRTSP::ScheduleParserHelper::airline\_code\_p(), AIRTSP::ScheduleParserHelper::airport\_p(), AIRTSP::ScheduleParserHelper::cabin\_code\_p(), AIRTSP::ScheduleParserHelper::class\_code\_list\_p(), AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::date, AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::date\_offset, AIRTSP::ScheduleParserHelper::day\_p(), AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::dow, AIRTSP::ScheduleParserHelper::dow\_p(), AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::family\_cabin\_details, AIRTSP::ScheduleParserHelper::family\_code\_p, AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::flight\_key, AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::flight\_number, AIRTSP::ScheduleParserHelper::flight\_number\_p(), AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::flight\_period, AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::flight\_period\_end, AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::flight\_period\_list, AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::full\_segment\_cabin\_details, AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::generic\_segment, AIRTSP::ScheduleParserHelper::hours\_p(), AIRTSP::ScheduleParserHelper::intl\_p, AIRTSP::ScheduleParserHelper::key\_p(), AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::leg, AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::leg\_cabin\_details, AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::leg\_details, AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::leg\_key, AIRTSP::ScheduleParserHelper::minutes\_p(), AIRTSP::ScheduleParserHelper::month\_p(), AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::not\_to\_be\_parsed,

AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::operating\_leg\_details, AIRTSP::ScheduleParserHelper::seconds\_p(), AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::segment\_cabin\_details, AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::segment\_key, AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::segment\_section, AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::specific\_segment\_list, AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::time, and AIRTSP::ScheduleParserHelper::year\_p().

### 10.20.3 Member Function Documentation

#### 10.20.3.1 `template<typename ScannerT> bsc::rule< ScannerT > const & AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::start () const`

Entry point of the parser.

Definition at line 701 of file ScheduleParserHelper.cpp.

References AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::flight\_period\_list.

### 10.20.4 Member Data Documentation

#### 10.20.4.1 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::flight_period_list`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition(), and AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::start().

#### 10.20.4.2 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::flight_period`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

#### 10.20.4.3 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::not_to_be_parsed`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

#### 10.20.4.4 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::flight_period_end`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**10.20.4.5** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::flight_key`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**10.20.4.6** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::airline_code`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**10.20.4.7** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::flight_number`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**10.20.4.8** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::date`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**10.20.4.9** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::dow`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**10.20.4.10** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::time`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**10.20.4.11** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::date_offset`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**10.20.4.12** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::leg`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**10.20.4.13** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::leg_key`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**10.20.4.14** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::operating - leg_details`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**10.20.4.15** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::leg_details`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**10.20.4.16** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::leg_cabin_details`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**10.20.4.17** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::segment_section`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**10.20.4.18** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::segment_key`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**10.20.4.19** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::full_segment_cabin_details`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**10.20.4.20** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::segment_cabin_details`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**10.20.4.21** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::full_family_cabin_details`

Definition at line 291 of file ScheduleParserHelper.hpp.

**10.20.4.22** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::family_cabin_details`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**10.20.4.23** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::generic_segment`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**10.20.4.24** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::specific_segment_list`

Definition at line 291 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

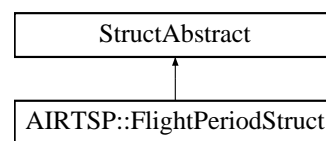
The documentation for this struct was generated from the following files:

- airtsp/command/ScheduleParserHelper.hpp
- airtsp/command/ScheduleParserHelper.cpp

## 10.21 AIRTSP::FlightPeriodStruct Struct Reference

```
#include <airtsp/bom/FlightPeriodStruct.hpp>
```

Inheritance diagram for AIRTSP::FlightPeriodStruct:



### Public Member Functions

- stdair::Date\_T [getDate](#) () const
- stdair::Duration\_T [getTime](#) () const
- const std::string [describe](#) () const
- void [addAirport](#) (const stdair::AirportCode\_T &)
- void [buildSegments](#) ()
- void [addSegmentCabin](#) (const [SegmentStruct](#) &, const [SegmentCabinStruct](#) &)
- void [addSegmentCabin](#) (const [SegmentCabinStruct](#) &)
- void [addFareFamily](#) (const [SegmentStruct](#) &, const [SegmentCabinStruct](#) &, const [FareFamilyStruct](#) &)
- void [addFareFamily](#) (const [SegmentCabinStruct](#) &, const [FareFamilyStruct](#) &)
- [FlightPeriodStruct](#) ()

### Public Attributes

- stdair::AirlineCode\_T [\\_airlineCode](#)
- stdair::FlightNumber\_T [\\_flightNumber](#)
- stdair::DatePeriod\_T [\\_dateRange](#)
- stdair::DoWStruct [\\_dow](#)
- [LegStructList\\_T](#) [\\_legList](#)
- [SegmentStructList\\_T](#) [\\_segmentList](#)
- bool [\\_legAlreadyDefined](#)
- [LegStruct](#) [\\_itLeg](#)
- [LegCabinStruct](#) [\\_itLegCabin](#)
- stdair::Date\_T [\\_dateRangeStart](#)
- stdair::Date\_T [\\_dateRangeEnd](#)
- unsigned int [\\_itYear](#)
- unsigned int [\\_itMonth](#)
- unsigned int [\\_itDay](#)
- int [\\_dateOffset](#)



- long [\\_itHours](#)
- long [\\_itMinutes](#)
- long [\\_itSeconds](#)
- [AirportList\\_T \\_airportList](#)
- [AirportOrderedList\\_T \\_airportOrderedList](#)
- bool [\\_areSegmentDefinitionsSpecific](#)
- [SegmentStruct \\_itSegment](#)
- [SegmentCabinStruct \\_itSegmentCabin](#)

### 10.21.1 Detailed Description

Utility Structure for the parsing of Flight-Period structures.

Definition at line 26 of file FlightPeriodStruct.hpp.

### 10.21.2 Constructor & Destructor Documentation

#### 10.21.2.1 AIRTSP::FlightPeriodStruct::FlightPeriodStruct ()

Constructor.

Definition at line 17 of file FlightPeriodStruct.cpp.

### 10.21.3 Member Function Documentation

#### 10.21.3.1 stdair::Date\_T AIRTSP::FlightPeriodStruct::getDate () const

Set the date from the staging details.

Definition at line 24 of file FlightPeriodStruct.cpp.

References [\\_itDay](#), [\\_itMonth](#), and [\\_itYear](#).

Referenced by [AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator\(\)\(\)](#), [AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator\(\)\(\)](#), and [AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator\(\)\(\)](#).

#### 10.21.3.2 stdair::Duration\_T AIRTSP::FlightPeriodStruct::getTime () const

Set the time from the staging details.

Definition at line 29 of file FlightPeriodStruct.cpp.

References [\\_itHours](#), [\\_itMinutes](#), and [\\_itSeconds](#).

Referenced by [AIRTSP::ScheduleParserHelper::storeElapsedTime::operator\(\)\(\)](#), [AIRTSP::ScheduleParserHelper::storeOffTime::operator\(\)\(\)](#), [AIRTSP::ScheduleParserHelper::storeBoardingTime::operator\(\)\(\)](#), and [AIRTSP::ScheduleParserHelper::storeBoardingTime::operator\(\)\(\)](#).

#### 10.21.3.3 const std::string AIRTSP::FlightPeriodStruct::describe () const

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

Definition at line 36 of file FlightPeriodStruct.cpp.

References [\\_airlineCode](#), [\\_dateRange](#), [\\_dow](#), [\\_flightNumber](#), [\\_legList](#), and [\\_segmentList](#).

Referenced by [AIRTSP::ScheduleParserHelper::doEndFlight::operator\(\)\(\)](#).

**10.21.3.4 void AIRTSP::FlightPeriodStruct::addAirport (const stdair::AirportCode\_T &)**

Add the given airport to the internal lists (if not already existing).

Definition at line 62 of file FlightPeriodStruct.cpp.

References `_airportList`, and `_airportOrderedList`.

Referenced by `AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator()()`, and `AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator()()`.

**10.21.3.5 void AIRTSP::FlightPeriodStruct::buildSegments ()**

Build the list of [SegmentStruct](#) objects.

Definition at line 78 of file FlightPeriodStruct.cpp.

References `_airportList`, `_airportOrderedList`, and `_segmentList`.

Referenced by `AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator()()`.

**10.21.3.6 void AIRTSP::FlightPeriodStruct::addSegmentCabin (const [SegmentStruct](#) &, const [SegmentCabinStruct](#) &)**

Add, to the Segment structure whose key corresponds to the given (board point, off point) pair, the specific segment cabin details (mainly, the list of the class codes).

Note that the Segment structure is retrieved from the internal list, already filled by a previous step (the [buildSegments\(\)](#) method).

Definition at line 111 of file FlightPeriodStruct.cpp.

References `AIRTSP::SegmentStruct::_boardingPoint`, `AIRTSP::SegmentStruct::_offPoint`, and `_segmentList`.

Referenced by `AIRTSP::ScheduleParserHelper::storeClasses::operator()()`.

**10.21.3.7 void AIRTSP::FlightPeriodStruct::addSegmentCabin (const [SegmentCabinStruct](#) &)**

Add, to all the Segment structures, the general segment cabin details (mainly, the list of the class codes).

Note that the Segment structures are stored within the internal list, already filled by a previous step (the [buildSegments\(\)](#) method).

Definition at line 149 of file FlightPeriodStruct.cpp.

References `_segmentList`.

**10.21.3.8 void AIRTSP::FlightPeriodStruct::addFareFamily (const [SegmentStruct](#) &, const [SegmentCabinStruct](#) &, const [FareFamilyStruct](#) &)**

Add, to the SegmentCabin structure whose key corresponds to the given cabin code, the specific segment fare family details (mainly, the list of the class codes).

Note that the SegmentCabin structure is retrieved from the internal list, already filled by a previous step (the [buildSegmentCabins\(\)](#) method).

Definition at line 162 of file FlightPeriodStruct.cpp.

References `AIRTSP::SegmentStruct::_boardingPoint`, `AIRTSP::SegmentCabinStruct::_cabinCode`, `AIRTSP::SegmentStruct::_offPoint`, and `_segmentList`.

Referenced by `AIRTSP::ScheduleParserHelper::storeFCClasses::operator()()`.

**10.21.3.9 void AIRTSP::FlightPeriodStruct::addFareFamily (const SegmentCabinStruct &, const FareFamilyStruct &)**

Add, to all the Segment structures, the general fare family sets (list of fare families).

Note that the SegmentCabin structures are stored within the internal list, already filled by a previous step (the buildSegmentCabins() method).

Definition at line 229 of file FlightPeriodStruct.cpp.

References AIRTSP::SegmentCabinStruct::\_cabinCode, and \_segmentList.

**10.21.4 Member Data Documentation****10.21.4.1 stdair::AirlineCode\_T AIRTSP::FlightPeriodStruct::\_airlineCode**

Definition at line 84 of file FlightPeriodStruct.hpp.

Referenced by describe(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

**10.21.4.2 stdair::FlightNumber\_T AIRTSP::FlightPeriodStruct::\_flightNumber**

Definition at line 85 of file FlightPeriodStruct.hpp.

Referenced by describe(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), and AIRTSP::ScheduleParserHelper::storeFlightNumber::operator().

**10.21.4.3 stdair::DatePeriod\_T AIRTSP::FlightPeriodStruct::\_dateRange**

Definition at line 86 of file FlightPeriodStruct.hpp.

Referenced by describe(), and AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator().

**10.21.4.4 stdair::DoWStruct AIRTSP::FlightPeriodStruct::\_dow**

Definition at line 87 of file FlightPeriodStruct.hpp.

Referenced by describe(), and AIRTSP::ScheduleParserHelper::storeDow::operator().

**10.21.4.5 LegStructList\_T AIRTSP::FlightPeriodStruct::\_legList**

Definition at line 88 of file FlightPeriodStruct.hpp.

Referenced by describe(), AIRTSP::ScheduleParserHelper::doEndFlight::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

**10.21.4.6 SegmentStructList\_T AIRTSP::FlightPeriodStruct::\_segmentList**

Definition at line 89 of file FlightPeriodStruct.hpp.

Referenced by addFareFamily(), addSegmentCabin(), buildSegments(), and describe().

**10.21.4.7 bool AIRTSP::FlightPeriodStruct::\_legAlreadyDefined**

Staging Leg (resp. Cabin) structure, gathering the result of the iteration on one leg (resp. cabin).

Definition at line 93 of file FlightPeriodStruct.hpp.

Referenced by AIRTSP::ScheduleParserHelper::doEndFlight::operator(), and AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator().

#### 10.21.4.8 LegStruct AIRTSP::FlightPeriodStruct::\_itLeg

Definition at line 94 of file FlightPeriodStruct.hpp.

Referenced by AIRTSP::ScheduleParserHelper::doEndFlight::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), and AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator().

#### 10.21.4.9 LegCabinStruct AIRTSP::FlightPeriodStruct::\_itLegCabin

Definition at line 95 of file FlightPeriodStruct.hpp.

Referenced by AIRTSP::ScheduleParserHelper::storeCapacity::operator(), and AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator().

#### 10.21.4.10 stdair::Date\_T AIRTSP::FlightPeriodStruct::\_dateRangeStart

Staging Date.

Definition at line 98 of file FlightPeriodStruct.hpp.

Referenced by AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), and AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator().

#### 10.21.4.11 stdair::Date\_T AIRTSP::FlightPeriodStruct::\_dateRangeEnd

Definition at line 99 of file FlightPeriodStruct.hpp.

Referenced by AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator().

#### 10.21.4.12 unsigned int AIRTSP::FlightPeriodStruct::\_itYear

Definition at line 100 of file FlightPeriodStruct.hpp.

Referenced by getDate().

#### 10.21.4.13 unsigned int AIRTSP::FlightPeriodStruct::\_itMonth

Definition at line 101 of file FlightPeriodStruct.hpp.

Referenced by getDate().

#### 10.21.4.14 unsigned int AIRTSP::FlightPeriodStruct::\_itDay

Definition at line 102 of file FlightPeriodStruct.hpp.

Referenced by getDate().

**10.21.4.15 int AIRTSP::FlightPeriodStruct::\_dateOffset**

Definition at line 103 of file FlightPeriodStruct.hpp.

Referenced by AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), and AIRTSP::ScheduleParserHelper::storeBoardingTime::operator().

**10.21.4.16 long AIRTSP::FlightPeriodStruct::\_itHours**

Staging Time.

Definition at line 106 of file FlightPeriodStruct.hpp.

Referenced by getTime().

**10.21.4.17 long AIRTSP::FlightPeriodStruct::\_itMinutes**

Definition at line 107 of file FlightPeriodStruct.hpp.

Referenced by getTime().

**10.21.4.18 long AIRTSP::FlightPeriodStruct::\_itSeconds**

Definition at line 108 of file FlightPeriodStruct.hpp.

Referenced by getTime(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), and AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator().

**10.21.4.19 AirportList\_T AIRTSP::FlightPeriodStruct::\_airportList**

Staging Airport List (helper to derive the list of Segment structures).

Definition at line 112 of file FlightPeriodStruct.hpp.

Referenced by addAirport(), buildSegments(), and AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator().

**10.21.4.20 AirportOrderedList\_T AIRTSP::FlightPeriodStruct::\_airportOrderedList**

Definition at line 113 of file FlightPeriodStruct.hpp.

Referenced by addAirport(), buildSegments(), and AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator().

**10.21.4.21 bool AIRTSP::FlightPeriodStruct::\_areSegmentDefinitionsSpecific**

Staging Segment-related attributes.

Definition at line 116 of file FlightPeriodStruct.hpp.

Referenced by AIRTSP::ScheduleParserHelper::storeFClasses::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), and AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator().

**10.21.4.22 SegmentStruct AIRTSP::FlightPeriodStruct::\_itSegment**

Definition at line 117 of file FlightPeriodStruct.hpp.

Referenced by AIRTSP::ScheduleParserHelper::storeFClasses::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), and AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator().

**10.21.4.23 SegmentCabinStruct AIRTSP::FlightPeriodStruct::\_itSegmentCabin**

Definition at line 118 of file FlightPeriodStruct.hpp.

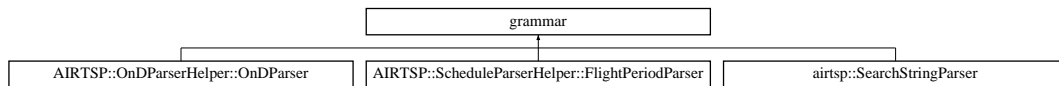
Referenced by AIRTSP::ScheduleParserHelper::storeFClasses::operator(), AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), and AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator().

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

- airtsp/bom/[FlightPeriodStruct.hpp](#)
- airtsp/bom/[FlightPeriodStruct.cpp](#)

**10.22 grammar Class Reference**

Inheritance diagram for grammar::



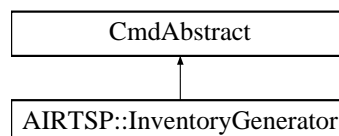
The documentation for this class was generated from the following files:

- airtsp/batches/[BookingRequestParser.cpp](#)
- airtsp/command/[ScheduleParserHelper.hpp](#)
- airtsp/command/[OnDParserHelper.hpp](#)

**10.23 AIRTSP::InventoryGenerator Class Reference**

```
#include <airtsp/command/InventoryGenerator.hpp>
```

Inheritance diagram for AIRTSP::InventoryGenerator::

**Friends**

- class [FlightPeriodFileParser](#)

- class [FFFlightPeriodFileParser](#)
- struct [ScheduleParserHelper::doEndFlight](#)
- class [ScheduleParser](#)

### 10.23.1 Detailed Description

Class handling the generation / instantiation of the Inventory BOM.

Definition at line 31 of file InventoryGenerator.hpp.

### 10.23.2 Friends And Related Function Documentation

#### 10.23.2.1 friend class [FlightPeriodFileParser](#) [friend]

Definition at line 35 of file InventoryGenerator.hpp.

#### 10.23.2.2 friend class [FFFlightPeriodFileParser](#) [friend]

Definition at line 36 of file InventoryGenerator.hpp.

#### 10.23.2.3 friend struct [ScheduleParserHelper::doEndFlight](#) [friend]

Definition at line 37 of file InventoryGenerator.hpp.

#### 10.23.2.4 friend class [ScheduleParser](#) [friend]

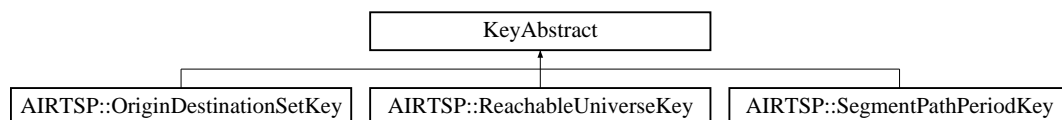
Definition at line 38 of file InventoryGenerator.hpp.

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

- airtsp/command/[InventoryGenerator.hpp](#)
- airtsp/command/[InventoryGenerator.cpp](#)

## 10.24 KeyAbstract Class Reference

Inheritance diagram for KeyAbstract::



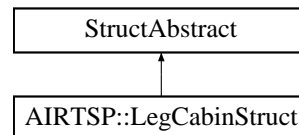
The documentation for this class was generated from the following files:

- airtsp/bom/[OriginDestinationSetKey.hpp](#)
- airtsp/bom/[ReachableUniverseKey.hpp](#)
- airtsp/bom/[SegmentPathPeriodKey.hpp](#)

## 10.25 AIRTSP::LegCabinStruct Struct Reference

```
#include <airtsp/bom/LegCabinStruct.hpp>
```

Inheritance diagram for AIRTSP::LegCabinStruct:



### Public Member Functions

- void [fill](#) (stdair::LegCabin &) const
- const std::string [describe](#) () const

### Public Attributes

- stdair::CabinCode\_T [\\_cabinCode](#)
- stdair::CabinCapacity\_T [\\_capacity](#)

### 10.25.1 Detailed Description

Utility Structure for the parsing of LegCabin details.

Definition at line 22 of file LegCabinStruct.hpp.

### 10.25.2 Member Function Documentation

#### 10.25.2.1 void AIRTSP::LegCabinStruct::fill (stdair::LegCabin &) const

Fill the LegCabin objects with the attributes of the [LegCabinStruct](#).

Definition at line 22 of file LegCabinStruct.cpp.

References [\\_capacity](#).

#### 10.25.2.2 const std::string AIRTSP::LegCabinStruct::describe () const

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

Definition at line 15 of file LegCabinStruct.cpp.

References [\\_cabinCode](#), and [\\_capacity](#).

### 10.25.3 Member Data Documentation

#### 10.25.3.1 stdair::CabinCode\_T AIRTSP::LegCabinStruct::\_cabinCode

Definition at line 24 of file LegCabinStruct.hpp.

Referenced by [describe\(\)](#), and [AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator\(\)\(\)](#).



### 10.25.3.2 stdair::CabinCapacity\_T AIRTSP::LegCabinStruct::\_capacity

Definition at line 25 of file LegCabinStruct.hpp.

Referenced by describe(), fill(), and AIRTSP::ScheduleParserHelper::storeCapacity::operator()().

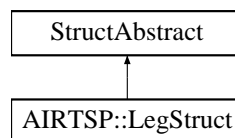
The documentation for this struct was generated from the following files:

- airtsp/bom/LegCabinStruct.hpp
- airtsp/bom/LegCabinStruct.cpp

## 10.26 AIRTSP::LegStruct Struct Reference

```
#include <airtsp/bom/LegStruct.hpp>
```

Inheritance diagram for AIRTSP::LegStruct:



### Public Member Functions

- void [fill](#) (const stdair::Date\_T &iRefDate, stdair::LegDate &) const
- const std::string [describe](#) () const
- [LegStruct](#) ()

### Public Attributes

- stdair::AirlineCode\_T [\\_airlineCode](#)
- stdair::FlightNumber\_T [\\_flightNumber](#)
- stdair::AirportCode\_T [\\_boardingPoint](#)
- stdair::DateOffset\_T [\\_boardingDateOffset](#)
- stdair::Duration\_T [\\_boardingTime](#)
- stdair::AirportCode\_T [\\_offPoint](#)
- stdair::DateOffset\_T [\\_offDateOffset](#)
- stdair::Duration\_T [\\_offTime](#)
- stdair::Duration\_T [\\_elapsed](#)
- [LegCabinStructList\\_T](#) [\\_cabinList](#)

### 10.26.1 Detailed Description

Utility Structure for the parsing of Leg structures.

Definition at line 24 of file LegStruct.hpp.

## 10.26.2 Constructor & Destructor Documentation

### 10.26.2.1 AIRTSP::LegStruct::LegStruct ()

Default Constructor.

Definition at line 16 of file LegStruct.cpp.

## 10.26.3 Member Function Documentation

### 10.26.3.1 void AIRTSP::LegStruct::fill (const stdair::Date\_T & iRefDate, stdair::LegDate &) const

Fill the LegDate objects with the attributes of the [LegStruct](#).

The given reference date corresponds to the date of the FlightDate. Indeed, each Leg gets date off-sets, when compared to that (reference) flight-date, both for the boarding date and for the off date.

Definition at line 48 of file LegStruct.cpp.

References `_airlineCode`, `_boardingDateOffset`, `_boardingTime`, `_elapsed`, `_flightNumber`, `_offDateOffset`, `_offPoint`, and `_offTime`.

### 10.26.3.2 const std::string AIRTSP::LegStruct::describe () const

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

Definition at line 22 of file LegStruct.cpp.

References `_boardingDateOffset`, `_boardingPoint`, `_boardingTime`, `_cabinList`, `_elapsed`, `_offDateOffset`, `_offPoint`, and `_offTime`.

## 10.26.4 Member Data Documentation

### 10.26.4.1 stdair::AirlineCode\_T AIRTSP::LegStruct::\_airlineCode

Definition at line 26 of file LegStruct.hpp.

Referenced by `fill()`, `AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator()()`, and `AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator()()`.

### 10.26.4.2 stdair::FlightNumber\_T AIRTSP::LegStruct::\_flightNumber

Definition at line 27 of file LegStruct.hpp.

Referenced by `fill()`, `AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator()()`, and `AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator()()`.

### 10.26.4.3 stdair::AirportCode\_T AIRTSP::LegStruct::\_boardingPoint

Definition at line 28 of file LegStruct.hpp.

Referenced by `describe()`, and `AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator()()`.

### 10.26.4.4 stdair::DateOffset\_T AIRTSP::LegStruct::\_boardingDateOffset

Definition at line 29 of file LegStruct.hpp.

Referenced by `describe()`, `fill()`, and `AIRTSP::ScheduleParserHelper::storeOffTime::operator()()`.

**10.26.4.5 stdair::Duration\_T AIRTSP::LegStruct::\_boardingTime**

Definition at line 30 of file LegStruct.hpp.

Referenced by describe(), fill(), and AIRTSP::ScheduleParserHelper::storeBoardingTime::operator()().

**10.26.4.6 stdair::AirportCode\_T AIRTSP::LegStruct::\_offPoint**

Definition at line 31 of file LegStruct.hpp.

Referenced by describe(), fill(), and AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator()().

**10.26.4.7 stdair::DateOffset\_T AIRTSP::LegStruct::\_offDateOffset**

Definition at line 32 of file LegStruct.hpp.

Referenced by describe(), AIRTSP::SegmentPeriodHelper::fill(), fill(), and AIRTSP::ScheduleParserHelper::storeElapsedTime::operator()().

**10.26.4.8 stdair::Duration\_T AIRTSP::LegStruct::\_offTime**

Definition at line 33 of file LegStruct.hpp.

Referenced by describe(), AIRTSP::SegmentPeriodHelper::fill(), fill(), and AIRTSP::ScheduleParserHelper::storeOffTime::operator()().

**10.26.4.9 stdair::Duration\_T AIRTSP::LegStruct::\_elapsed**

Definition at line 34 of file LegStruct.hpp.

Referenced by describe(), fill(), and AIRTSP::ScheduleParserHelper::storeElapsedTime::operator()().

**10.26.4.10 LegCabinStructList\_T AIRTSP::LegStruct::\_cabinList**

Definition at line 35 of file LegStruct.hpp.

Referenced by describe(), AIRTSP::ScheduleParserHelper::doEndFlight::operator()(), AIRTSP::ScheduleParserHelper::storeCapacity::operator()(), and AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator()().

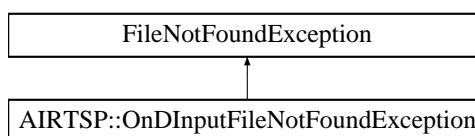
The documentation for this struct was generated from the following files:

- airtsp/bom/LegStruct.hpp
- airtsp/bom/LegStruct.cpp

**10.27 AIRTSP::OnInputFileNotFoundException Class Reference**

```
#include <airtsp/AIRTSP_Types.hpp>
```

Inheritance diagram for AIRTSP::OnInputFileNotFoundException::



## Public Member Functions

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

### 10.27.1 Detailed Description

The O&D input file cannot be retrieved.

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

### 10.27.2 Constructor & Destructor Documentation

#### 10.27.2.1 AIRTSP::OnDInputFileNotFoundException::OnDInputFileNotFoundException (const std::string &iWhat) [inline]

Constructor.

Definition at line 40 of file AIRTSP\_Types.hpp.

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

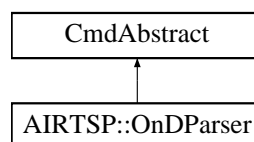
- airtsp/[AIRTSP\\_Types.hpp](#)

## 10.28 AIRTSP::OnDParser Class Reference

Class wrapping the parser entry point.

```
#include <airtsp/command/OnDParser.hpp>
```

Inheritance diagram for AIRTSP::OnDParser::



## Static Public Member Functions

- static void [generateOnDPeriods](#) (const stdair::ODFilePath &, stdair::BomRoot &)

### 10.28.1 Detailed Description

Class wrapping the parser entry point.

Definition at line 24 of file OnDParser.hpp.

### 10.28.2 Member Function Documentation

#### 10.28.2.1 void AIRTSP::OnDParser::generateOnDPeriods (const stdair::ODFilePath &, stdair::BomRoot &) [static]

Parse the CSV file describing the O&D.

**Parameters:**

**const** stdair::ODFilePath& The file-name of the CSV-formatted fare input file and the container.

Definition at line 17 of file OnDParser.cpp.

References AIRTSP::OnDPeriodFileParser::generateOnDPeriods().

Referenced by AIRTSP::AIRTSP\_Service::parseAndLoad().

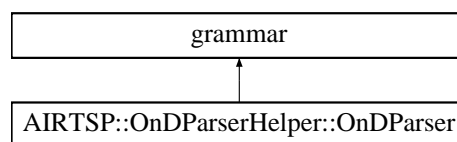
The documentation for this class was generated from the following files:

- airtsp/command/[OnDParser.hpp](#)
- airtsp/command/[OnDParser.cpp](#)

**10.29 AIRTSP::OnDParserHelper::OnDParser Struct Reference**

```
#include <airtsp/command/OnDParserHelper.hpp>
```

Inheritance diagram for AIRTSP::OnDParserHelper::OnDParser::

**Public Member Functions**

- [OnDParser](#) (stdair::BomRoot &, [OnDPeriodStruct](#) &)

**Public Attributes**

- stdair::BomRoot & [\\_bomRoot](#)
- [OnDPeriodStruct](#) & [\\_onDPeriod](#)

**Classes**

- struct [definition](#)

**10.29.1 Detailed Description**

Grammar for the FareRule parser.

Definition at line 127 of file OnDParserHelper.hpp.

**10.29.2 Constructor & Destructor Documentation****10.29.2.1 AIRTSP::OnDParserHelper::OnDParser::OnDParser (stdair::BomRoot &, [OnDPeriodStruct](#) &)**

Definition at line 261 of file OnDParserHelper.cpp.

### 10.29.3 Member Data Documentation

#### 10.29.3.1 stdair::BomRoot& [AIRTSP::OnDParserHelper::OnDParser::\\_bomRoot](#)

Definition at line 145 of file OnDParserHelper.hpp.

Referenced by [AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::definition\(\)](#).

#### 10.29.3.2 [OnDPeriodStruct](#)& [AIRTSP::OnDParserHelper::OnDParser::\\_onDPeriod](#)

Definition at line 146 of file OnDParserHelper.hpp.

Referenced by [AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::definition\(\)](#).

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

- [airtsp/command/OnDParserHelper.hpp](#)
- [airtsp/command/OnDParserHelper.cpp](#)

## 10.30 AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT > Struct Template Reference

```
#include <airtsp/command/OnDParserHelper.hpp>
```

### Public Member Functions

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

### Public Attributes

- [boost::spirit::classic::rule< ScannerT > ond\\_list](#)
- [boost::spirit::classic::rule< ScannerT > ond](#)
- [boost::spirit::classic::rule< ScannerT > segment](#)
- [boost::spirit::classic::rule< ScannerT > ond\\_key](#)
- [boost::spirit::classic::rule< ScannerT > ond\\_end](#)
- [boost::spirit::classic::rule< ScannerT > date](#)
- [boost::spirit::classic::rule< ScannerT > time](#)

### 10.30.1 Detailed Description

```
template<typename ScannerT> struct AIRTSP::OnDParserHelper::OnDParser::definition<
ScannerT >
```

Definition at line 133 of file OnDParserHelper.hpp.

### 10.30.2 Constructor & Destructor Documentation

#### 10.30.2.1 [template<typename ScannerT> AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::definition](#) ([OnDParser](#) const &self)

Definition at line 267 of file OnDParserHelper.cpp.

References AIRTSP::OnDParserHelper::OnDParser::\_bomRoot, AIRTSP::OnDParserHelper::OnDParser::\_onDPeriod, AIRTSP::OnDParserHelper::airline\_code\_p(), AIRTSP::OnDParserHelper::airport\_p(), AIRTSP::OnDParserHelper::class\_code\_p(), AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::date, AIRTSP::OnDParserHelper::day\_p(), AIRTSP::OnDParserHelper::hours\_p(), AIRTSP::OnDParserHelper::minutes\_p(), AIRTSP::OnDParserHelper::month\_p(), AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::ond, AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::ond\_end, AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::ond\_key, AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::ond\_list, AIRTSP::OnDParserHelper::seconds\_p(), AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::segment, AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::time, and AIRTSP::OnDParserHelper::year\_p().

### 10.30.3 Member Function Documentation

#### 10.30.3.1 `template<typename ScannerT> boost::spirit::classic::rule< ScannerT > const & AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::start () const`

Entry point of the parser.

Definition at line 330 of file OnDParserHelper.cpp.

References AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::ond\_list.

### 10.30.4 Member Data Documentation

#### 10.30.4.1 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::ond_list`

Definition at line 137 of file OnDParserHelper.hpp.

Referenced by AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::definition(), and AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::start().

#### 10.30.4.2 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::ond`

Definition at line 137 of file OnDParserHelper.hpp.

Referenced by AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::definition().

#### 10.30.4.3 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::segment`

Definition at line 137 of file OnDParserHelper.hpp.

Referenced by AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::definition().

#### 10.30.4.4 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::ond_key`

Definition at line 137 of file OnDParserHelper.hpp.

Referenced by AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::definition().

#### 10.30.4.5 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::ond_end`

Definition at line 137 of file OnDParserHelper.hpp.

Referenced by AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::definition().

#### 10.30.4.6 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::date`

Definition at line 137 of file OnDParserHelper.hpp.

Referenced by AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::definition().

#### 10.30.4.7 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::time`

Definition at line 137 of file OnDParserHelper.hpp.

Referenced by AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >::definition().

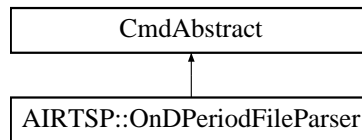
The documentation for this struct was generated from the following files:

- airtsp/command/OnDParserHelper.hpp
- airtsp/command/OnDParserHelper.cpp

## 10.31 AIRTSP::OnDPeriodFileParser Class Reference

```
#include <airtsp/command/OnDParserHelper.hpp>
```

Inheritance diagram for AIRTSP::OnDPeriodFileParser::



### Public Member Functions

- [OnDPeriodFileParser](#) (const stdair::Filename\_T &iFilename, stdair::BomRoot &ioBomRoot)
- bool [generateOnDPeriods](#) ()

#### 10.31.1 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 161 of file OnDParserHelper.hpp.

#### 10.31.2 Constructor & Destructor Documentation

##### 10.31.2.1 AIRTSP::OnDPeriodFileParser::OnDPeriodFileParser (const stdair::Filename\_T &iFilename, stdair::BomRoot &ioBomRoot)



Constructor.

Definition at line 342 of file OnDParserHelper.cpp.

### 10.31.3 Member Function Documentation

#### 10.31.3.1 bool AIRTSP::OnDPeriodFileParser::generateOnDPeriods ()

Parse the input file and generate the O&D-Periods.

Definition at line 378 of file OnDParserHelper.cpp.

Referenced by AIRTSP::OnDParser::generateOnDPeriods().

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

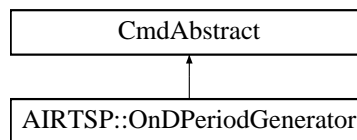
- airtsp/command/[OnDParserHelper.hpp](#)
- airtsp/command/[OnDParserHelper.cpp](#)

## 10.32 AIRTSP::OnDPeriodGenerator Class Reference

Class handling the generation / instantiation of the O&D-Period BOM.

```
#include <airtsp/command/OnDPeriodGenerator.hpp>
```

Inheritance diagram for AIRTSP::OnDPeriodGenerator::



### Friends

- class [OnDPeriodFileParser](#)
- struct [OnDParserHelper::doEndOnD](#)
- class [OnDParser](#)

### 10.32.1 Detailed Description

Class handling the generation / instantiation of the O&D-Period BOM.

Definition at line 29 of file OnDPeriodGenerator.hpp.

### 10.32.2 Friends And Related Function Documentation

#### 10.32.2.1 friend class [OnDPeriodFileParser](#) [friend]

Only the following class may use methods of [OnDPeriodGenerator](#). Indeed, as those methods build the BOM, it is not good to expose them publicly.

Definition at line 35 of file OnDPeriodGenerator.hpp.

**10.32.2.2 friend struct [OnDParserHelper::doEndOnD](#)** [friend]

Definition at line 36 of file OnDPeriodGenerator.hpp.

**10.32.2.3 friend class [OnDParser](#)** [friend]

Definition at line 37 of file OnDPeriodGenerator.hpp.

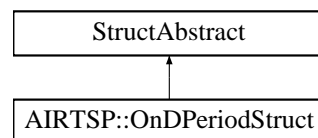
The documentation for this class was generated from the following files:

- airtsp/command/[OnDPeriodGenerator.hpp](#)
- airtsp/command/[OnDPeriodGenerator.cpp](#)

**10.33 AIRTSP::OnDPeriodStruct Struct Reference**

```
#include <airtsp/bom/OnDPeriodStruct.hpp>
```

Inheritance diagram for AIRTSP::OnDPeriodStruct::

**Public Member Functions**

- const stdair::AirlineCode\_T & [getFirstAirlineCode](#) () const
- stdair::Date\_T [getDate](#) () const
- stdair::Duration\_T [getTime](#) () const
- const std::string [describe](#) () const
- const std::string [describeTSKey](#) () const
- [OnDPeriodStruct](#) ()

**Public Attributes**

- stdair::AirportCode\_T [\\_origin](#)
- stdair::AirportCode\_T [\\_destination](#)
- stdair::DatePeriod\_T [\\_datePeriod](#)
- stdair::Duration\_T [\\_timeRangeStart](#)
- stdair::Duration\_T [\\_timeRangeEnd](#)
- stdair::NbOfAirlines\_T [\\_nbOfAirlines](#)
- stdair::AirlineCode\_T [\\_airlineCode](#)
- stdair::ClassCode\_T [\\_classCode](#)
- stdair::AirlineCodeList\_T [\\_airlineCodeList](#)
- stdair::ClassCodeList\_T [\\_classCodeList](#)
- stdair::Date\_T [\\_dateRangeStart](#)
- stdair::Date\_T [\\_dateRangeEnd](#)
- unsigned int [\\_itYear](#)
- unsigned int [\\_itMonth](#)
- unsigned int [\\_itDay](#)

- long [\\_itHours](#)
- long [\\_itMinutes](#)
- long [\\_itSeconds](#)

### 10.33.1 Detailed Description

Utility Structure for the parsing of FareRule structures.

Definition at line 16 of file OnDPeriodStruct.hpp.

### 10.33.2 Constructor & Destructor Documentation

#### 10.33.2.1 AIRTSP::OnDPeriodStruct::OnDPeriodStruct ()

Default constructor.

Definition at line 17 of file OnDPeriodStruct.cpp.

### 10.33.3 Member Function Documentation

#### 10.33.3.1 const stdair::AirlineCode\_T & AIRTSP::OnDPeriodStruct::getFirstAirlineCode () const

Get the first airline code.

Definition at line 64 of file OnDPeriodStruct.cpp.

References [\\_airlineCodeList](#).

#### 10.33.3.2 stdair::Date\_T AIRTSP::OnDPeriodStruct::getDate () const

Get the date from the staging details.

Definition at line 28 of file OnDPeriodStruct.cpp.

References [\\_itDay](#), [\\_itMonth](#), and [\\_itYear](#).

Referenced by [AIRTSP::OnDParserHelper::storeDateRangeEnd::operator\(\)](#), and [AIRTSP::OnDParserHelper::storeDateRangeStart::operator\(\)](#).

#### 10.33.3.3 stdair::Duration\_T AIRTSP::OnDPeriodStruct::getTime () const

Get the time from the staging details.

Definition at line 33 of file OnDPeriodStruct.cpp.

References [\\_itHours](#), [\\_itMinutes](#), and [\\_itSeconds](#).

Referenced by [AIRTSP::OnDParserHelper::storeEndRangeTime::operator\(\)](#), and [AIRTSP::OnDParserHelper::storeStartRangeTime::operator\(\)](#).

#### 10.33.3.4 const std::string AIRTSP::OnDPeriodStruct::describe () const

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

Definition at line 40 of file OnDPeriodStruct.cpp.

References [\\_airlineCode](#), [\\_classCode](#), [\\_datePeriod](#), [\\_destination](#), [\\_origin](#), [\\_timeRangeEnd](#), and [\\_timeRangeStart](#).

#### 10.33.3.5 `const std::string AIRTSP::OnDPeriodStruct::describeTSKey () const`

Give a short description of the key required in the travel solution object to differentiate fare rule structures.

Definition at line 55 of file OnDPeriodStruct.cpp.

References `_airlineCode`, `_classCode`, `_destination`, and `_origin`.

### 10.33.4 Member Data Documentation

#### 10.33.4.1 `stdair::AirportCode_T AIRTSP::OnDPeriodStruct::_origin`

Definition at line 42 of file OnDPeriodStruct.hpp.

Referenced by `describe()`, `describeTSKey()`, and `AIRTSP::OnDParserHelper::storeOrigin::operator()`.

#### 10.33.4.2 `stdair::AirportCode_T AIRTSP::OnDPeriodStruct::_destination`

Definition at line 43 of file OnDPeriodStruct.hpp.

Referenced by `describe()`, `describeTSKey()`, and `AIRTSP::OnDParserHelper::storeDestination::operator()`.

#### 10.33.4.3 `stdair::DatePeriod_T AIRTSP::OnDPeriodStruct::_datePeriod`

Definition at line 44 of file OnDPeriodStruct.hpp.

Referenced by `describe()`, and `AIRTSP::OnDParserHelper::storeDateRangeEnd::operator()`.

#### 10.33.4.4 `stdair::Duration_T AIRTSP::OnDPeriodStruct::_timeRangeStart`

Definition at line 45 of file OnDPeriodStruct.hpp.

Referenced by `describe()`, and `AIRTSP::OnDParserHelper::storeStartRangeTime::operator()`.

#### 10.33.4.5 `stdair::Duration_T AIRTSP::OnDPeriodStruct::_timeRangeEnd`

Definition at line 46 of file OnDPeriodStruct.hpp.

Referenced by `describe()`, and `AIRTSP::OnDParserHelper::storeEndRangeTime::operator()`.

#### 10.33.4.6 `stdair::NbOfAirlines_T AIRTSP::OnDPeriodStruct::_nbOfAirlines`

Definition at line 47 of file OnDPeriodStruct.hpp.

Referenced by `AIRTSP::OnDParserHelper::storeAirlineCode::operator()`, and `AIRTSP::OnDParserHelper::storeOrigin::operator()`.

#### 10.33.4.7 `stdair::AirlineCode_T AIRTSP::OnDPeriodStruct::_airlineCode`

Definition at line 48 of file OnDPeriodStruct.hpp.

Referenced by `describe()`, `describeTSKey()`, `AIRTSP::OnDParserHelper::storeAirlineCode::operator()`, and `AIRTSP::OnDParserHelper::storeOrigin::operator()`.

#### 10.33.4.8 `stdair::ClassCode_T AIRTSP::OnDPeriodStruct::_classCode`

Definition at line 49 of file OnDPeriodStruct.hpp.

Referenced by describe(), describeTSKey(), AIRTSP::OnDParserHelper::storeClassCode::operator>(), and AIRTSP::OnDParserHelper::storeOrigin::operator().

#### 10.33.4.9 stdair::AirlineCodeList\_T AIRTSP::OnDPeriodStruct::\_airlineCodeList

Definition at line 50 of file OnDPeriodStruct.hpp.

Referenced by getFirstAirlineCode(), AIRTSP::OnDParserHelper::storeAirlineCode::operator(), and AIRTSP::OnDParserHelper::storeOrigin::operator().

#### 10.33.4.10 stdair::ClassCodeList\_T AIRTSP::OnDPeriodStruct::\_classCodeList

Definition at line 51 of file OnDPeriodStruct.hpp.

Referenced by AIRTSP::OnDParserHelper::storeClassCode::operator(), and AIRTSP::OnDParserHelper::storeOrigin::operator().

#### 10.33.4.11 stdair::Date\_T AIRTSP::OnDPeriodStruct::\_dateRangeStart

Staging Date.

Definition at line 54 of file OnDPeriodStruct.hpp.

Referenced by AIRTSP::OnDParserHelper::storeDateRangeEnd::operator(), and AIRTSP::OnDParserHelper::storeDateRangeStart::operator().

#### 10.33.4.12 stdair::Date\_T AIRTSP::OnDPeriodStruct::\_dateRangeEnd

Definition at line 55 of file OnDPeriodStruct.hpp.

Referenced by AIRTSP::OnDParserHelper::storeDateRangeEnd::operator().

#### 10.33.4.13 unsigned int AIRTSP::OnDPeriodStruct::\_itYear

Definition at line 56 of file OnDPeriodStruct.hpp.

Referenced by getDate().

#### 10.33.4.14 unsigned int AIRTSP::OnDPeriodStruct::\_itMonth

Definition at line 57 of file OnDPeriodStruct.hpp.

Referenced by getDate().

#### 10.33.4.15 unsigned int AIRTSP::OnDPeriodStruct::\_itDay

Definition at line 58 of file OnDPeriodStruct.hpp.

Referenced by getDate().

#### 10.33.4.16 long AIRTSP::OnDPeriodStruct::\_itHours

Staging Time.

Definition at line 61 of file OnDPeriodStruct.hpp.

Referenced by getTime().

**10.33.4.17 long AIRTSP::OnDPeriodStruct::\_itMinutes**

Definition at line 62 of file OnDPeriodStruct.hpp.

Referenced by getTime().

**10.33.4.18 long AIRTSP::OnDPeriodStruct::\_itSeconds**

Definition at line 63 of file OnDPeriodStruct.hpp.

Referenced by getTime(), AIRTSP::OnDParserHelper::storeEndRangeTime::operator>(), AIRTSP::OnDParserHelper::storeStartRangeTime::operator>(), AIRTSP::OnDParserHelper::storeDateRangeEnd::operator>(), and AIRTSP::OnDParserHelper::storeDateRangeStart::operator().

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

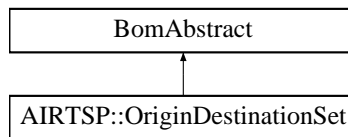
- airtsp/bom/OnDPeriodStruct.hpp
- airtsp/bom/OnDPeriodStruct.cpp

**10.34 AIRTSP::OriginDestinationSet Class Reference**

Class representing a simple sub-network.

```
#include <airtsp/bom/OriginDestinationSet.hpp>
```

Inheritance diagram for AIRTSP::OriginDestinationSet:

**Public Types**

- typedef [OriginDestinationSetKey](#) Key\_T

**Public Member Functions**

- const [Key\\_T](#) & [getKey](#) () const
- const stdair::AirportCode\_T & [getDestination](#) () const
- stdair::BomAbstract \*const [getParent](#) () const
- const stdair::HolderMap\_T & [getHolderMap](#) () const
- void [toStream](#) (std::ostream &ioOut) const
- void [fromStream](#) (std::istream &ioIn)
- std::string [toString](#) () const
- const std::string [describeKey](#) () const
- template<class Archive> void [serialize](#) (Archive &ar, const unsigned int iFileVersion)

**Protected Member Functions**

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

### Protected Attributes

- [Key\\_T \\_key](#)
- [stdair::BomAbstract \\* \\_parent](#)
- [stdair::HolderMap\\_T \\_holderMap](#)

### Friends

- class [stdair::FacBom](#)
- class [stdair::FacBomManager](#)
- class [boost::serialization::access](#)

#### 10.34.1 Detailed Description

Class representing a simple sub-network.

That simple sub-network is made of a set of segments ([SegmentPathPeriod](#) objects), from the origin airport specified within [ReachableUniverse](#) (parent object) to the destination specified in the [OriginDestinationSetKey](#) object.

Each segment (composing that [OriginDestinationSet](#) object) corresponds to an actual travel solution from the origin to the destination, that is, a path that a traveller can take with actual scheduled flights.

Definition at line 44 of file [OriginDestinationSet.hpp](#).

#### 10.34.2 Member Typedef Documentation

##### 10.34.2.1 typedef [OriginDestinationSetKey](#) AIRTSP::OriginDestinationSet::Key\_T

Definition allowing to retrieve the associated BOM key type.

Definition at line 57 of file [OriginDestinationSet.hpp](#).

#### 10.34.3 Constructor & Destructor Documentation

##### 10.34.3.1 AIRTSP::OriginDestinationSet::OriginDestinationSet (const [Key\\_T](#) &) [protected]

Main constructor.

Definition at line 31 of file [OriginDestinationSet.cpp](#).

##### 10.34.3.2 AIRTSP::OriginDestinationSet::~~OriginDestinationSet () [protected]

Destructor.

Definition at line 36 of file [OriginDestinationSet.cpp](#).

#### 10.34.4 Member Function Documentation

##### 10.34.4.1 const [Key\\_T](#)& AIRTSP::OriginDestinationSet::getKey () const [inline]

Get the primary key (destination airport).

Definition at line 65 of file [OriginDestinationSet.hpp](#).

References `_key`.

**10.34.4.2** `const stdair::AirportCode_T& AIRTSP::OriginDestinationSet::getDestination () const [inline]`

Get the destination airport (i.e., the primary key).

Definition at line 72 of file `OriginDestinationSet.hpp`.

References `_key`, and `AIRTSP::OriginDestinationSetKey::getOffPoint()`.

**10.34.4.3** `stdair::BomAbstract* const AIRTSP::OriginDestinationSet::getParent () const [inline]`

Get the parent (i.e., [ReachableUniverse](#)) object.

Definition at line 79 of file `OriginDestinationSet.hpp`.

References `_parent`.

**10.34.4.4** `const stdair::HolderMap_T& AIRTSP::OriginDestinationSet::getHolderMap () const [inline]`

Get the map of children holders ([SegmentPathPeriod](#) objects).

Definition at line 86 of file `OriginDestinationSet.hpp`.

References `_holderMap`.

**10.34.4.5** `void AIRTSP::OriginDestinationSet::toStream (std::ostream & ioOut) const [inline]`

Dump a Business Object into an output stream.

#### Parameters:

*ostream&* the output stream.

Definition at line 98 of file `OriginDestinationSet.hpp`.

References `toString()`.

**10.34.4.6** `void AIRTSP::OriginDestinationSet::fromStream (std::istream & ioIn) [inline]`

Read a Business Object from an input stream.

#### Parameters:

*istream&* the input stream.

Definition at line 107 of file `OriginDestinationSet.hpp`.

**10.34.4.7** `std::string AIRTSP::OriginDestinationSet::toString () const`

Get the serialised version of the Business Object.

Definition at line 40 of file `OriginDestinationSet.cpp`.



References `_key`, and `AIRTSP::OriginDestinationSetKey::toString()`.

Referenced by `toStream()`.

#### 10.34.4.8 `const std::string AIRTSP::OriginDestinationSet::describeKey () const` `[inline]`

Get a string describing the key.

Definition at line 118 of file `OriginDestinationSet.hpp`.

References `_key`, and `AIRTSP::OriginDestinationSetKey::toString()`.

#### 10.34.4.9 `template<class Archive> void AIRTSP::OriginDestinationSet::serialize (Archive & ar, const unsigned int iFileVersion)`

Serialisation.

Definition at line 62 of file `OriginDestinationSet.cpp`.

References `_key`.

### 10.34.5 Friends And Related Function Documentation

#### 10.34.5.1 `friend class stdair::FacBom` `[friend]`

Friend classes.

Definition at line 48 of file `OriginDestinationSet.hpp`.

#### 10.34.5.2 `friend class stdair::FacBomManager` `[friend]`

Definition at line 49 of file `OriginDestinationSet.hpp`.

#### 10.34.5.3 `friend class boost::serialization::access` `[friend]`

Definition at line 50 of file `OriginDestinationSet.hpp`.

### 10.34.6 Member Data Documentation

#### 10.34.6.1 `Key_T AIRTSP::OriginDestinationSet::_key` `[protected]`

Primary key (destination airport code).

Definition at line 168 of file `OriginDestinationSet.hpp`.

Referenced by `describeKey()`, `getDestination()`, `getKey()`, `serialize()`, and `toString()`.

#### 10.34.6.2 `stdair::BomAbstract* AIRTSP::OriginDestinationSet::_parent` `[protected]`

Pointer on the parent ([ReachableUniverse](#)) object.

Definition at line 173 of file `OriginDestinationSet.hpp`.

Referenced by `getParent()`.

**10.34.6.3 stdair::HolderMap\_T AIRTSP::OriginDestinationSet::\_holderMap** [protected]

Map holding the children ([SegmentPathPeriod](#) objects).

Definition at line 178 of file [OriginDestinationSet.hpp](#).

Referenced by [getHolderMap\(\)](#).

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

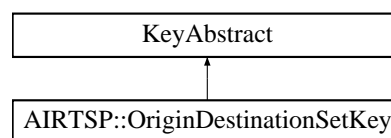
- [airtsp/bom/OriginDestinationSet.hpp](#)
- [airtsp/bom/OriginDestinationSet.cpp](#)

**10.35 AIRTSP::OriginDestinationSetKey Struct Reference**

Structure representing the key of a sub-network.

```
#include <airtsp/bom/OriginDestinationSetKey.hpp>
```

Inheritance diagram for AIRTSP::OriginDestinationSetKey::

**Public Member Functions**

- [OriginDestinationSetKey](#) (const stdair::AirportCode\_T &iDestination)
- [OriginDestinationSetKey](#) (const [OriginDestinationSetKey](#) &)
- [~OriginDestinationSetKey](#) ()
- const stdair::AirportCode\_T & [getOffPoint](#) () const
- void [toStream](#) (std::ostream &ioOut) const
- void [fromStream](#) (std::istream &ioIn)
- const std::string [toString](#) () const
- template<class Archive> void [serialize](#) (Archive &ar, const unsigned int iFileVersion)

**Friends**

- class [boost::serialization::access](#)

**10.35.1 Detailed Description**

Structure representing the key of a sub-network.

As the origin airport code is already part of the [ReachableUniverse](#) (parent) class, that key is only made of the destination airport code.

Definition at line 30 of file [OriginDestinationSetKey.hpp](#).

## 10.35.2 Constructor & Destructor Documentation

### 10.35.2.1 AIRTSP::OriginDestinationSetKey::OriginDestinationSetKey (const stdair::AirportCode\_T & *iDestination*)

Constructor.

Definition at line 26 of file OriginDestinationSetKey.cpp.

### 10.35.2.2 AIRTSP::OriginDestinationSetKey::OriginDestinationSetKey (const [OriginDestinationSetKey](#) &)

Copy constructor.

Definition at line 32 of file OriginDestinationSetKey.cpp.

### 10.35.2.3 AIRTSP::OriginDestinationSetKey::~~OriginDestinationSetKey ()

Destructor.

Definition at line 37 of file OriginDestinationSetKey.cpp.

## 10.35.3 Member Function Documentation

### 10.35.3.1 const stdair::AirportCode\_T& AIRTSP::OriginDestinationSetKey::getOffPoint () const [inline]

Get the destination airport.

Definition at line 62 of file OriginDestinationSetKey.hpp.

Referenced by AIRTSP::OriginDestinationSet::getDestination().

### 10.35.3.2 void AIRTSP::OriginDestinationSetKey::toStream (std::ostream & *ioOut*) const

Dump a Business Object Key into an output stream.

#### Parameters:

*ostream&* the output stream.

Definition at line 41 of file OriginDestinationSetKey.cpp.

References toString().

### 10.35.3.3 void AIRTSP::OriginDestinationSetKey::fromStream (std::istream & *ioIn*)

Read a Business Object Key from an input stream.

#### Parameters:

*istream&* the input stream.

Definition at line 46 of file OriginDestinationSetKey.cpp.

**10.35.3.4 const std::string AIRTSP::OriginDestinationSetKey::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-date.

Definition at line 50 of file OriginDestinationSetKey.cpp.

Referenced by AIRTSP::OriginDestinationSet::describeKey(), toStream(), and AIRTSP::OriginDestinationSet::toString().

**10.35.3.5 template<class Archive> void AIRTSP::OriginDestinationSetKey::serialize (Archive & ar, const unsigned int iFileVersion)**

Serialisation.

Definition at line 72 of file OriginDestinationSetKey.cpp.

**10.35.4 Friends And Related Function Documentation****10.35.4.1 friend class boost::serialization::access [friend]**

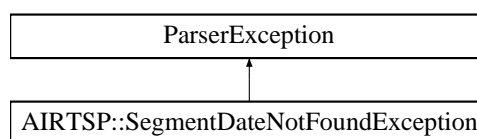
Definition at line 31 of file OriginDestinationSetKey.hpp.

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

- [airtsp/bom/OriginDestinationSetKey.hpp](#)
- [airtsp/bom/OriginDestinationSetKey.cpp](#)

**10.36 ParseException Class Reference**

Inheritance diagram for ParseException::



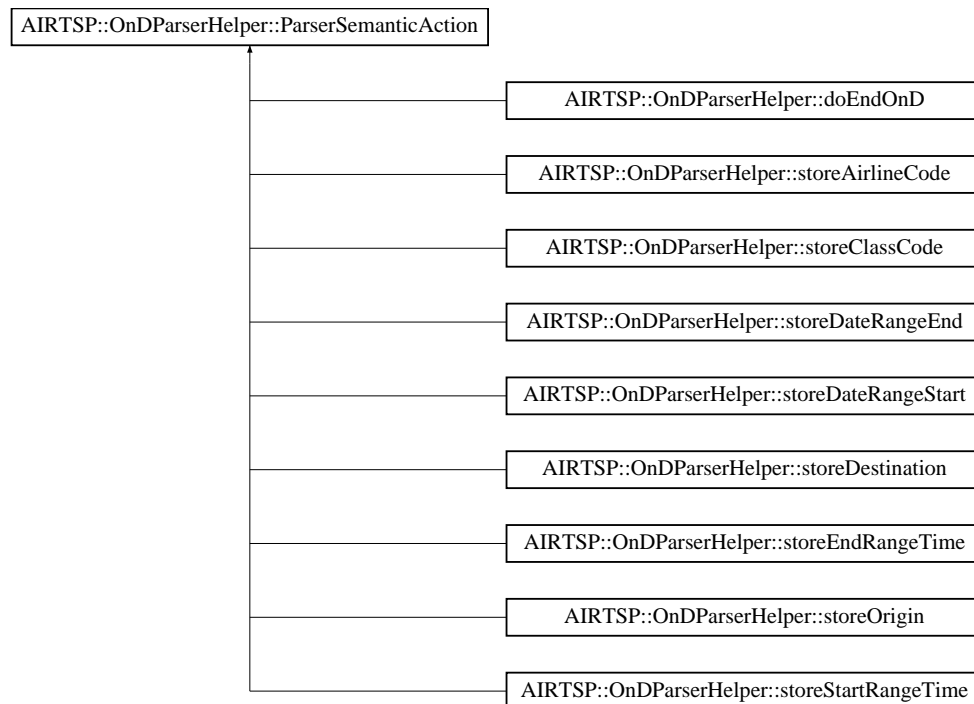
The documentation for this class was generated from the following file:

- [airtsp/AIRTSP\\_Types.hpp](#)

**10.37 AIRTSP::OnDParserHelper::ParserSemanticAction Struct Reference**

```
#include <airtsp/command/OnDParserHelper.hpp>
```

Inheritance diagram for AIRTSP::OnDParserHelper::ParserSemanticAction::



## Public Member Functions

- [ParserSemanticAction](#) ([OnDPeriodStruct](#) &)

## Public Attributes

- [OnDPeriodStruct](#) & [\\_onDPeriod](#)

### 10.37.1 Detailed Description

Generic Semantic Action (Actor / Functor) for the Schedule Parser.

Definition at line 34 of file OnDParserHelper.hpp.

### 10.37.2 Constructor & Destructor Documentation

#### 10.37.2.1 AIRTSP::OnDParserHelper::ParserSemanticAction::ParserSemanticAction ([OnDPeriodStruct](#) &)

Actor Constructor.

Definition at line 25 of file OnDParserHelper.cpp.

### 10.37.3 Member Data Documentation

#### 10.37.3.1 [OnDPeriodStruct](#)& [AIRTSP::OnDParserHelper::ParserSemanticAction::\\_onDPeriod](#)

Actor Context.

Definition at line 38 of file OnDParserHelper.hpp.

Referenced by AIRTSP::OnDParserHelper::doEndOnD::operator(), AIRTSP::OnDParserHelper::storeClassCode::operator(), AIRTSP::OnDParserHelper::storeAirlineCode::operator(), AIRTSP::OnDParserHelper::storeEndRangeTime::operator(), AIRTSP::OnDParserHelper::storeStartRangeTime::operator(), AIRTSP::OnDParserHelper::storeDateRangeEnd::operator(), AIRTSP::OnDParserHelper::storeDateRangeStart::operator(), AIRTSP::OnDParserHelper::storeDestination::operator(), and AIRTSP::OnDParserHelper::storeOrigin::operator().

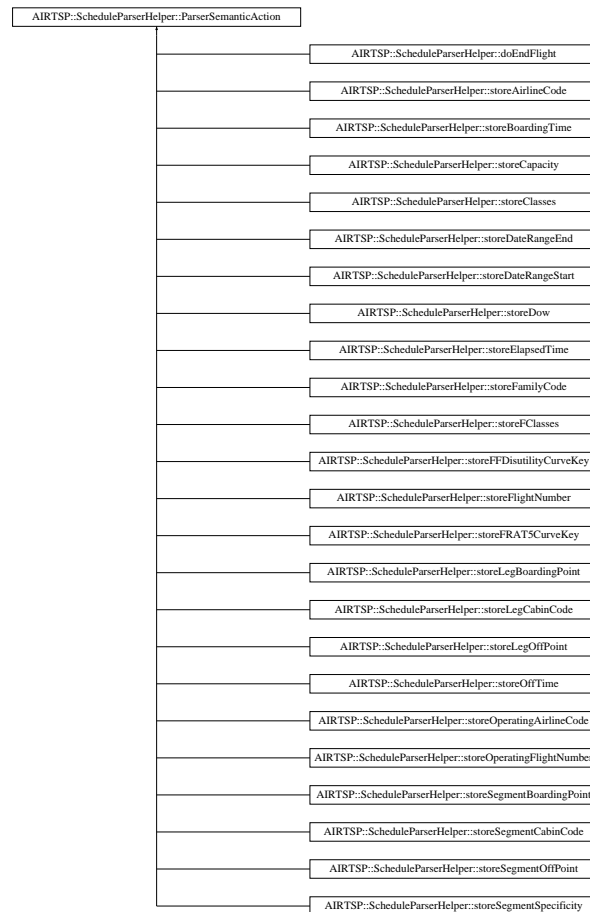
The documentation for this struct was generated from the following files:

- [airtsp/command/OnDParserHelper.hpp](#)
- [airtsp/command/OnDParserHelper.cpp](#)

## 10.38 AIRTSP::ScheduleParserHelper::ParserSemanticAction Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::ParserSemanticAction:



### Public Member Functions

- [ParserSemanticAction](#) ([FlightPeriodStruct](#) &)

**Public Attributes**

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

**10.38.1 Detailed Description**

Generic Semantic Action (Actor / Functor) for the Schedule Parser.

Definition at line 29 of file `ScheduleParserHelper.hpp`.

**10.38.2 Constructor & Destructor Documentation****10.38.2.1 AIRTSP::ScheduleParserHelper::ParserSemanticAction::ParserSemanticAction([FlightPeriodStruct](#) &)**

Actor Constructor.

Definition at line 26 of file `ScheduleParserHelper.cpp`.

**10.38.3 Member Data Documentation****10.38.3.1 [FlightPeriodStruct](#)& AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod**

Actor Context.

Definition at line 33 of file `ScheduleParserHelper.hpp`.

Referenced by `AIRTSP::ScheduleParserHelper::doEndFlight::operator()`, `AIRTSP::ScheduleParserHelper::storeFClasses::operator()`, `AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator()`, `AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator()`, `AIRTSP::ScheduleParserHelper::storeFamilyCode::operator()`, `AIRTSP::ScheduleParserHelper::storeClasses::operator()`, `AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator()`, `AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator()`, `AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator()`, `AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator()`, `AIRTSP::ScheduleParserHelper::storeCapacity::operator()`, `AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator()`, `AIRTSP::ScheduleParserHelper::storeElapsedTime::operator()`, `AIRTSP::ScheduleParserHelper::storeOffTime::operator()`, `AIRTSP::ScheduleParserHelper::storeBoardingTime::operator()`, `AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator()`, `AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator()`, `AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator()`, `AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator()`, `AIRTSP::ScheduleParserHelper::storeDow::operator()`, `AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator()`, `AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator()`, `AIRTSP::ScheduleParserHelper::storeFlightNumber::operator()`, and `AIRTSP::ScheduleParserHelper::storeAirlineCode::operator()`.

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

- `airtsp/command/ScheduleParserHelper.hpp`
- `airtsp/command/ScheduleParserHelper.cpp`

**10.39 airtsp::Passenger\_T Struct Reference**

```
#include <airtsp/batches/BookingRequestParser.hpp>
```

### Public Types

- [ADULT](#) = 0
- [CHILD](#)
- [PET](#)
- [LAST\\_VALUE](#)
- enum [PassengerType\\_T](#) { [ADULT](#) = 0, [CHILD](#), [PET](#), [LAST\\_VALUE](#) }

### Public Member Functions

- [Passenger\\_T](#) ()
- void [display](#) () const

### Public Attributes

- [PassengerType\\_T \\_type](#)
- unsigned short [\\_number](#)

### Static Public Attributes

- static const std::string [\\_labels](#) [[LAST\\_VALUE](#)]

## 10.39.1 Detailed Description

Passenger.

Definition at line 71 of file BookingRequestParser.hpp.

## 10.39.2 Member Enumeration Documentation

### 10.39.2.1 enum [airtsp::Passenger\\_T::PassengerType\\_T](#)

Enumerator:

*[ADULT](#)*

*[CHILD](#)*

*[PET](#)*

*[LAST\\_VALUE](#)*

Definition at line 73 of file BookingRequestParser.hpp.

## 10.39.3 Constructor & Destructor Documentation

### 10.39.3.1 [airtsp::Passenger\\_T::Passenger\\_T](#) () [[inline](#)]

Constructor.

Definition at line 78 of file BookingRequestParser.hpp.



### 10.39.4 Member Function Documentation

#### 10.39.4.1 void airtsp::Passenger\_T::display () const [inline]

Definition at line 80 of file BookingRequestParser.hpp.

References `_labels`, `_number`, and `_type`.

### 10.39.5 Member Data Documentation

#### 10.39.5.1 const std::string airtsp::Passenger\_T::\_labels [static]

Initial value:

```
{ "Adult", "Child", "Pet" }
```

Passenger type labels.

Definition at line 74 of file BookingRequestParser.hpp.

Referenced by `display()`.

#### 10.39.5.2 PassengerType\_T airtsp::Passenger\_T::\_type

Definition at line 75 of file BookingRequestParser.hpp.

Referenced by `display()`, `airtsp::store_pet_passenger_type::operator()`, `airtsp::store_child_passenger_type::operator()`, and `airtsp::store_adult_passenger_type::operator()`.

#### 10.39.5.3 unsigned short airtsp::Passenger\_T::\_number

Definition at line 76 of file BookingRequestParser.hpp.

Referenced by `display()`, and `airtsp::store_passenger_number::operator()`.

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

- `airtsp/batches/BookingRequestParser.hpp`

## 10.40 airtsp::Place\_T Struct Reference

```
#include <airtsp/batches/BookingRequestParser.hpp>
```

### Public Member Functions

- `Place_T ()`
- void `display () const`

### Public Attributes

- std::string `_name`
- std::string `_code`

### 10.40.1 Detailed Description

Place.

Definition at line 11 of file BookingRequestParser.hpp.

### 10.40.2 Constructor & Destructor Documentation

#### 10.40.2.1 airtsp::Place\_T::Place\_T () [inline]

Constructor.

Definition at line 16 of file BookingRequestParser.hpp.

### 10.40.3 Member Function Documentation

#### 10.40.3.1 void airtsp::Place\_T::display () const [inline]

Definition at line 18 of file BookingRequestParser.hpp.

References `_code`, and `_name`.

Referenced by `airtsp::SearchString_T::display()`.

### 10.40.4 Member Data Documentation

#### 10.40.4.1 std::string airtsp::Place\_T::\_name

Definition at line 13 of file BookingRequestParser.hpp.

Referenced by `display()`, and `airtsp::store_place_element::operator()()`.

#### 10.40.4.2 std::string airtsp::Place\_T::\_code

Definition at line 14 of file BookingRequestParser.hpp.

Referenced by `display()`.

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

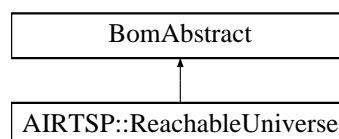
- [airtsp/batches/BookingRequestParser.hpp](#)

## 10.41 AIRTSP::ReachableUniverse Class Reference

Class representing the root of the schedule-related BOM tree.

```
#include <airtsp/bom/ReachableUniverse.hpp>
```

Inheritance diagram for AIRTSP::ReachableUniverse::



## Public Types

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

## Public Member Functions

- const [Key\\_T](#) & [getKey](#) () const
- const stdair::AirportCode\_T & [getOrigin](#) () const
- stdair::BomAbstract \*const [getParent](#) () const
- const stdair::HolderMap\_T & [getHolderMap](#) () const
- const [SegmentPathPeriodListList\\_T](#) & [getSegmentPathPeriodListList](#) () const
- void [toStream](#) (std::ostream &ioOut) const
- void [fromStream](#) (std::istream &ioIn)
- std::string [toString](#) () const
- const std::string [describeKey](#) () const
- template<class Archive> void [serialize](#) (Archive &ar, const unsigned int iFileVersion)

## Protected Member Functions

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

## Protected Attributes

- [Key\\_T \\_key](#)
- stdair::BomAbstract \* [\\_parent](#)
- stdair::HolderMap\_T [\\_holderMap](#)
- [SegmentPathPeriodListList\\_T \\_segmentPathPeriodListList](#)

## Friends

- class [stdair::FacBom](#)
- class [stdair::FacBomManager](#)
- class [SegmentPathGenerator](#)
- class [boost::serialization::access](#)

### 10.41.1 Detailed Description

Class representing the root of the schedule-related BOM tree.

It is the pending, in the schedule universe, of the stdair::Inventory class. It corresponds to all the destinations, which can be reached from a given geographical point. That latter is an airport for now, and its key (airport code) is specified by the [ReachableUniverseKey](#) object.

Definition at line 41 of file ReachableUniverse.hpp.

### 10.41.2 Member Typedef Documentation

#### 10.41.2.1 typedef [ReachableUniverseKey](#) [AIRTSP::ReachableUniverse::Key\\_T](#)

Definition allowing to retrieve the associated BOM key type.

Definition at line 55 of file ReachableUniverse.hpp.

### 10.41.3 Constructor & Destructor Documentation

#### 10.41.3.1 AIRTSP::ReachableUniverse::ReachableUniverse (const [Key\\_T](#) &) [protected]

Main constructor.

Definition at line 32 of file ReachableUniverse.cpp.

#### 10.41.3.2 AIRTSP::ReachableUniverse::~~ReachableUniverse () [protected]

Destructor.

Definition at line 37 of file ReachableUniverse.cpp.

### 10.41.4 Member Function Documentation

#### 10.41.4.1 const [Key\\_T](#)& AIRTSP::ReachableUniverse::getKey () const [inline]

Get the universe key (airport code representing the departure point of the "reachable universe").

Definition at line 63 of file ReachableUniverse.hpp.

References `_key`.

#### 10.41.4.2 const stdair::AirportCode\_T& AIRTSP::ReachableUniverse::getOrigin () const [inline]

Get the (origin) airport (i.e., the primary key).

Definition at line 70 of file ReachableUniverse.hpp.

References `_key`, and `AIRTSP::ReachableUniverseKey::getBoardingPoint()`.

#### 10.41.4.3 stdair::BomAbstract\* const AIRTSP::ReachableUniverse::getParent () const [inline]

Get the parent (i.e., the BomRoot) object.

Definition at line 77 of file ReachableUniverse.hpp.

References `_parent`.

#### 10.41.4.4 const stdair::HolderMap\_T& AIRTSP::ReachableUniverse::getHolderMap () const [inline]

Get the map of children holders ([OriginDestinationSet](#) objects).

Definition at line 84 of file ReachableUniverse.hpp.

References `_holderMap`.

#### 10.41.4.5 const [SegmentPathPeriodListList\\_T](#)& AIRTSP::ReachableUniverse::getSegmentPathPeriodListList () const [inline]

Get the vector of SegmentPathPeriodLightList objects.

Definition at line 91 of file ReachableUniverse.hpp.

References `_segmentPathPeriodListList`.

**10.41.4.6 void AIRTSP::ReachableUniverse::toStream (std::ostream & *ioOut*) const** [inline]

Dump a Business Object into an output stream.

**Parameters:**

*ostream&* the output stream.

Definition at line 103 of file ReachableUniverse.hpp.

References toString().

**10.41.4.7 void AIRTSP::ReachableUniverse::fromStream (std::istream & *ioIn*)** [inline]

Read a Business Object from an input stream.

**Parameters:**

*istream&* the input stream.

Definition at line 112 of file ReachableUniverse.hpp.

**10.41.4.8 std::string AIRTSP::ReachableUniverse::toString () const**

Get the serialised version of the Business Object.

Definition at line 41 of file ReachableUniverse.cpp.

References \_key, and AIRTSP::ReachableUniverseKey::toString().

Referenced by AIRTSP::BomDisplay::csvDisplay(), and toStream().

**10.41.4.9 const std::string AIRTSP::ReachableUniverse::describeKey () const** [inline]

Get a string describing the key.

Definition at line 123 of file ReachableUniverse.hpp.

References \_key, and AIRTSP::ReachableUniverseKey::toString().

**10.41.4.10 template<class Archive> void AIRTSP::ReachableUniverse::serialize (Archive & *ar*, const unsigned int *iFileVersion*)**

Serialisation.

Definition at line 63 of file ReachableUniverse.cpp.

References \_key.

**10.41.5 Friends And Related Function Documentation****10.41.5.1 friend class stdair::FacBom** [friend]

Friend classes.

Definition at line 45 of file ReachableUniverse.hpp.

**10.41.5.2 friend class stdair::FacBomManager** [friend]

Definition at line 46 of file ReachableUniverse.hpp.

**10.41.5.3 friend class [SegmentPathGenerator](#)** [friend]

Definition at line 47 of file ReachableUniverse.hpp.

**10.41.5.4 friend class boost::serialization::access** [friend]

Definition at line 48 of file ReachableUniverse.hpp.

**10.41.6 Member Data Documentation****10.41.6.1 [Key\\_T AIRTSP::ReachableUniverse::\\_key](#)** [protected]

Primary key (origin airport code).

Definition at line 174 of file ReachableUniverse.hpp.

Referenced by describeKey(), getKey(), getOrigin(), serialize(), and toString().

**10.41.6.2 stdair::BomAbstract\* [AIRTSP::ReachableUniverse::\\_parent](#)** [protected]

Pointer on the parent (BomRoot) object.

Definition at line 179 of file ReachableUniverse.hpp.

Referenced by getParent().

**10.41.6.3 stdair::HolderMap\_T [AIRTSP::ReachableUniverse::\\_holderMap](#)** [protected]

Map holding the children ([OriginDestinationSet](#) objects).

Definition at line 184 of file ReachableUniverse.hpp.

Referenced by getHolderMap().

**10.41.6.4 [SegmentPathPeriodListList\\_T AIRTSP::ReachableUniverse::\\_segmentPathPeriodListList](#)** [protected]

The list (actually, a vector) of lists of SegmentPathPeriods, used solely for the construction of the main list of SegmentPathPeriods within the ReachableUniverseStructure.

Definition at line 191 of file ReachableUniverse.hpp.

Referenced by getSegmentPathPeriodListList().

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

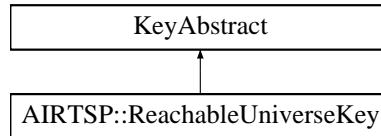
- airtsp/bom/[ReachableUniverse.hpp](#)
- airtsp/bom/[ReachableUniverse.cpp](#)

**10.42 AIRTSP::ReachableUniverseKey Struct Reference**

Structure representing the key of the schedule-related BOM tree root.

```
#include <airtsp/bom/ReachableUniverseKey.hpp>
```

Inheritance diagram for AIRTSP::ReachableUniverseKey::



### Public Member Functions

- [ReachableUniverseKey](#) (const stdair::AirportCode\_T &iOrigin)
- [ReachableUniverseKey](#) (const [ReachableUniverseKey](#) &)
- [~ReachableUniverseKey](#) ()
- const stdair::AirportCode\_T & [getBoardingPoint](#) () const
- void [toStream](#) (std::ostream &ioOut) const
- void [fromStream](#) (std::istream &ioIn)
- const std::string [toString](#) () const
- template<class Archive> void [serialize](#) (Archive &ar, const unsigned int iFileVersion)

### Friends

- class [boost::serialization::access](#)

#### 10.42.1 Detailed Description

Structure representing the key of the schedule-related BOM tree root.

The [ReachableUniverse](#) is the pending, in the schedule universe, of the stdair::Inventory class. It corresponds to all the destinations which can be reached from a given geographical point. That latter is an airport for now, and the present structure specifies its key (i.e., airport code).

Definition at line 33 of file [ReachableUniverseKey.hpp](#).

#### 10.42.2 Constructor & Destructor Documentation

##### 10.42.2.1 AIRTSP::ReachableUniverseKey::ReachableUniverseKey (const stdair::AirportCode\_T &iOrigin)

Constructor.

Definition at line 32 of file [ReachableUniverseKey.cpp](#).

##### 10.42.2.2 AIRTSP::ReachableUniverseKey::ReachableUniverseKey (const [ReachableUniverseKey](#) &)

Copy constructor.

Definition at line 26 of file [ReachableUniverseKey.cpp](#).

### 10.42.2.3 AIRTSP::ReachableUniverseKey::~~ReachableUniverseKey ()

Destructor.

Definition at line 37 of file ReachableUniverseKey.cpp.

## 10.42.3 Member Function Documentation

### 10.42.3.1 const std::air::AirportCode\_T& AIRTSP::ReachableUniverseKey::getBoardingPoint () const [inline]

Get the origin airport (from which the remaining universe may be reached).

Definition at line 66 of file ReachableUniverseKey.hpp.

Referenced by AIRTSP::ReachableUniverse::getOrigin().

### 10.42.3.2 void AIRTSP::ReachableUniverseKey::toStream (std::ostream & ioOut) const

Dump a Business Object Key into an output stream.

#### Parameters:

*ostream&* the output stream.

Definition at line 41 of file ReachableUniverseKey.cpp.

References toString().

### 10.42.3.3 void AIRTSP::ReachableUniverseKey::fromStream (std::istream & ioIn)

Read a Business Object Key from an input stream.

#### Parameters:

*istream&* the input stream.

Definition at line 46 of file ReachableUniverseKey.cpp.

### 10.42.3.4 const std::string AIRTSP::ReachableUniverseKey::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-date.

Definition at line 50 of file ReachableUniverseKey.cpp.

Referenced by AIRTSP::ReachableUniverse::describeKey(), toStream(), and AIRTSP::ReachableUniverse::toString().

### 10.42.3.5 template<class Archive> void AIRTSP::ReachableUniverseKey::serialize (Archive & ar, const unsigned int iFileVersion)

Serialisation.

Definition at line 72 of file ReachableUniverseKey.cpp.



### 10.42.4 Friends And Related Function Documentation

#### 10.42.4.1 friend class boost::serialization::access [friend]

Definition at line 34 of file ReachableUniverseKey.hpp.

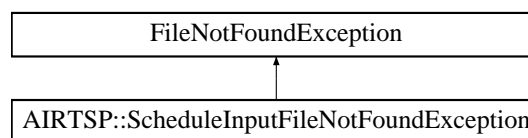
The documentation for this struct was generated from the following files:

- [airtsp/bom/ReachableUniverseKey.hpp](#)
- [airtsp/bom/ReachableUniverseKey.cpp](#)

## 10.43 AIRTSP::ScheduleInputFileNotFoundException Class Reference

```
#include <airtsp/AIRTSP_Types.hpp>
```

Inheritance diagram for AIRTSP::ScheduleInputFileNotFoundException::



### Public Member Functions

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

#### 10.43.1 Detailed Description

The schedule input file cannot be retrieved.

Definition at line 47 of file AIRTSP\_Types.hpp.

#### 10.43.2 Constructor & Destructor Documentation

##### 10.43.2.1 AIRTSP::ScheduleInputFileNotFoundException::ScheduleInputFileNotFoundException (const std::string &iWhat) [inline]

Constructor.

Definition at line 53 of file AIRTSP\_Types.hpp.

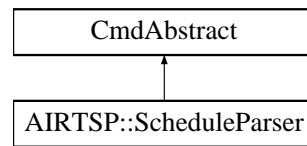
The documentation for this class was generated from the following file:

- [airtsp/AIRTSP\\_Types.hpp](#)

## 10.44 AIRTSP::ScheduleParser Class Reference

```
#include <airtsp/command/ScheduleParser.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParser::



### Static Public Member Functions

- static void [generateInventories](#) (const stdair::ScheduleFilePath &, stdair::BomRoot &)

#### 10.44.1 Detailed Description

Class wrapping the parser entry point.

Definition at line 22 of file ScheduleParser.hpp.

#### 10.44.2 Member Function Documentation

##### 10.44.2.1 void AIRTSP::ScheduleParser::generateInventories (const stdair::ScheduleFilePath &, stdair::BomRoot &) [static]

Parse the CSV file describing the airline schedules for the simulator, and generates the inventories accordingly.

#### Parameters:

- const stdair::ScheduleFilePath&** The file-name of the CSV-formatted schedule input file.
- stdair::BomRoot&** Root of the BOM tree.

Definition at line 20 of file ScheduleParser.cpp.

References AIRTSP::FlightPeriodFileParser::generateInventories().

Referenced by AIRTSP::AIRTSP\_Service::parseAndLoad().

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

- airtsp/command/[ScheduleParser.hpp](#)
- airtsp/command/[ScheduleParser.cpp](#)

## 10.45 airtsp::SearchString\_T Struct Reference

```
#include <airtsp/batches/BookingRequestParser.hpp>
```

### Public Member Functions

- [SearchString\\_T](#) ()
- void [display](#) () const

## Public Attributes

- [PlaceList\\_T \\_placeList](#)
- [DateList\\_T \\_dateList](#)
- [AirlineList\\_T \\_airlineList](#)
- [PassengerList\\_T \\_passengerList](#)
- [Place\\_T \\_tmpPlace](#)
- [Date\\_T \\_tmpDate](#)
- [Airline\\_T \\_tmpAirline](#)
- [Passenger\\_T \\_tmpPassenger](#)

### 10.45.1 Detailed Description

Search string.

Definition at line 94 of file BookingRequestParser.hpp.

### 10.45.2 Constructor & Destructor Documentation

#### 10.45.2.1 airtsp::SearchString\_T::SearchString\_T () [inline]

Constructor.

Definition at line 102 of file BookingRequestParser.hpp.

### 10.45.3 Member Function Documentation

#### 10.45.3.1 void airtsp::SearchString\_T::display () const [inline]

Definition at line 105 of file BookingRequestParser.hpp.

References [\\_airlineList](#), [\\_dateList](#), [\\_passengerList](#), [\\_placeList](#), [\\_tmpPlace](#), and [airtsp::Place\\_T::display\(\)](#).

### 10.45.4 Member Data Documentation

#### 10.45.4.1 [PlaceList\\_T airtsp::SearchString\\_T::\\_placeList](#)

Definition at line 96 of file BookingRequestParser.hpp.

Referenced by [display\(\)](#).

#### 10.45.4.2 [DateList\\_T airtsp::SearchString\\_T::\\_dateList](#)

Definition at line 97 of file BookingRequestParser.hpp.

Referenced by [display\(\)](#), and [airtsp::store\\_date::operator\(\)\(\)](#).

#### 10.45.4.3 [AirlineList\\_T airtsp::SearchString\\_T::\\_airlineList](#)

Definition at line 98 of file BookingRequestParser.hpp.

Referenced by [display\(\)](#), [airtsp::store\\_airline\\_name::operator\(\)\(\)](#), and [airtsp::store\\_airline\\_code::operator\(\)\(\)](#).

**10.45.4.4 PassengerList\_T airtsp::SearchString\_T::\_passengerList**

Definition at line 99 of file BookingRequestParser.hpp.

Referenced by `display()`, `airtsp::store_pet_passenger_type::operator()`, `airtsp::store_child_passenger_type::operator()`, and `airtsp::store_adult_passenger_type::operator()`.

**10.45.4.5 Place\_T airtsp::SearchString\_T::\_tmpPlace**

Definition at line 137 of file BookingRequestParser.hpp.

Referenced by `display()`, and `airtsp::store_place_element::operator()`.

**10.45.4.6 Date\_T airtsp::SearchString\_T::\_tmpDate**

Definition at line 138 of file BookingRequestParser.hpp.

Referenced by `airtsp::store_date::operator()`.

**10.45.4.7 Airline\_T airtsp::SearchString\_T::\_tmpAirline**

Definition at line 139 of file BookingRequestParser.hpp.

Referenced by `airtsp::store_airline_name::operator()`, `airtsp::store_airline_code::operator()`, and `airtsp::store_airline_sign::operator()`.

**10.45.4.8 Passenger\_T airtsp::SearchString\_T::\_tmpPassenger**

Definition at line 140 of file BookingRequestParser.hpp.

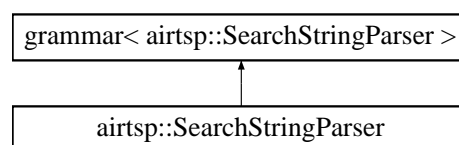
Referenced by `airtsp::store_pet_passenger_type::operator()`, `airtsp::store_child_passenger_type::operator()`, `airtsp::store_adult_passenger_type::operator()`, and `airtsp::store_passenger_number::operator()`.

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

- [airtsp/batches/BookingRequestParser.hpp](#)

**10.46 airtsp::SearchStringParser Struct Reference**

Inheritance diagram for `airtsp::SearchStringParser`:

**Public Member Functions**

- [SearchStringParser](#) ([SearchString\\_T](#) &ioSearchString)

### Public Attributes

- [SearchString\\_T](#) & [\\_searchString](#)

### Classes

- struct [definition](#)

#### 10.46.1 Detailed Description

Grammar for the search string parser.

Definition at line 251 of file BookingRequestParser.cpp.

#### 10.46.2 Constructor & Destructor Documentation

##### 10.46.2.1 airtsp::SearchStringParser::SearchStringParser ([SearchString\\_T](#) & *ioSearchString*) [inline]

Definition at line 254 of file BookingRequestParser.cpp.

#### 10.46.3 Member Data Documentation

##### 10.46.3.1 [SearchString\\_T](#) & [airtsp::SearchStringParser::\\_searchString](#)

Definition at line 369 of file BookingRequestParser.cpp.

Referenced by `airtsp::SearchStringParser::definition< ScannerT >::definition()`.

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

- `airtsp/batches/BookingRequestParser.cpp`

## 10.47 airtsp::SearchStringParser::definition< ScannerT > Struct Template Reference

### Public Member Functions

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

### Public Attributes

- `boost::spirit::classic::rule< ScannerT > search\_string`
- `boost::spirit::classic::rule< ScannerT > places`
- `boost::spirit::classic::rule< ScannerT > place\_element`
- `boost::spirit::classic::rule< ScannerT > dates`
- `boost::spirit::classic::rule< ScannerT > date`
- `boost::spirit::classic::rule< ScannerT > month`
- `boost::spirit::classic::rule< ScannerT > day`
- `boost::spirit::classic::rule< ScannerT > year`

- `boost::spirit::classic::rule< ScannerT >` [preferred\\_airlines](#)
- `boost::spirit::classic::rule< ScannerT >` [airline\\_code](#)
- `boost::spirit::classic::rule< ScannerT >` [airline\\_name](#)
- `boost::spirit::classic::rule< ScannerT >` [passengers](#)
- `boost::spirit::classic::rule< ScannerT >` [passenger\\_number](#)
- `boost::spirit::classic::rule< ScannerT >` [passenger\\_type](#)
- `boost::spirit::classic::rule< ScannerT >` [passenger\\_adult\\_type](#)
- `boost::spirit::classic::rule< ScannerT >` [passenger\\_child\\_type](#)
- `boost::spirit::classic::rule< ScannerT >` [passenger\\_pet\\_type](#)

### 10.47.1 Detailed Description

`template<typename ScannerT> struct airtsp::SearchStringParser::definition< ScannerT >`

Definition at line 259 of file `BookingRequestParser.cpp`.

### 10.47.2 Constructor & Destructor Documentation

**10.47.2.1** `template<typename ScannerT> airtsp::SearchStringParser::definition< ScannerT >::definition (SearchStringParser const & self)` `[inline]`

Definition at line 260 of file `BookingRequestParser.cpp`.

References `airtsp::SearchStringParser::_searchString`, `airtsp::SearchStringParser::definition< ScannerT >::airline_code`, `airtsp::SearchStringParser::definition< ScannerT >::airline_name`, `airtsp::SearchStringParser::definition< ScannerT >::date`, `airtsp::SearchStringParser::definition< ScannerT >::dates`, `airtsp::SearchStringParser::definition< ScannerT >::day`, `airtsp::SearchStringParser::definition< ScannerT >::month`, `airtsp::SearchStringParser::definition< ScannerT >::passenger_adult_type`, `airtsp::SearchStringParser::definition< ScannerT >::passenger_child_type`, `airtsp::SearchStringParser::definition< ScannerT >::passenger_number`, `airtsp::SearchStringParser::definition< ScannerT >::passenger_pet_type`, `airtsp::SearchStringParser::definition< ScannerT >::passenger_type`, `airtsp::SearchStringParser::definition< ScannerT >::passengers`, `airtsp::SearchStringParser::definition< ScannerT >::place_element`, `airtsp::SearchStringParser::definition< ScannerT >::places`, `airtsp::SearchStringParser::definition< ScannerT >::preferred_airlines`, `airtsp::SearchStringParser::definition< ScannerT >::search_string`, `airtsp::uint1_2_p`, `airtsp::uint1_p`, `airtsp::uint2_p`, `airtsp::uint4_p`, and `airtsp::SearchStringParser::definition< ScannerT >::year`.

### 10.47.3 Member Function Documentation

**10.47.3.1** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> const& airtsp::SearchStringParser::definition< ScannerT >::start () const` `[inline]`

Definition at line 366 of file `BookingRequestParser.cpp`.

References `airtsp::SearchStringParser::definition< ScannerT >::search_string`.

### 10.47.4 Member Data Documentation

**10.47.4.1** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> airtsp::SearchStringParser::definition< ScannerT >::search\_string`

Definition at line 360 of file `BookingRequestParser.cpp`.

Referenced by airtsp::SearchStringParser::definition< ScannerT >::definition(), and airtsp::SearchStringParser::definition< ScannerT >::start().

#### 10.47.4.2 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> airtsp::SearchStringParser::definition< ScannerT >::places`

Definition at line 360 of file BookingRequestParser.cpp.

Referenced by airtsp::SearchStringParser::definition< ScannerT >::definition().

#### 10.47.4.3 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> airtsp::SearchStringParser::definition< ScannerT >::place_element`

Definition at line 360 of file BookingRequestParser.cpp.

Referenced by airtsp::SearchStringParser::definition< ScannerT >::definition().

#### 10.47.4.4 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> airtsp::SearchStringParser::definition< ScannerT >::dates`

Definition at line 360 of file BookingRequestParser.cpp.

Referenced by airtsp::SearchStringParser::definition< ScannerT >::definition().

#### 10.47.4.5 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> airtsp::SearchStringParser::definition< ScannerT >::date`

Definition at line 360 of file BookingRequestParser.cpp.

Referenced by airtsp::SearchStringParser::definition< ScannerT >::definition().

#### 10.47.4.6 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> airtsp::SearchStringParser::definition< ScannerT >::month`

Definition at line 360 of file BookingRequestParser.cpp.

Referenced by airtsp::SearchStringParser::definition< ScannerT >::definition().

#### 10.47.4.7 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> airtsp::SearchStringParser::definition< ScannerT >::day`

Definition at line 360 of file BookingRequestParser.cpp.

Referenced by airtsp::SearchStringParser::definition< ScannerT >::definition().

#### 10.47.4.8 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> airtsp::SearchStringParser::definition< ScannerT >::year`

Definition at line 360 of file BookingRequestParser.cpp.

Referenced by airtsp::SearchStringParser::definition< ScannerT >::definition().

#### 10.47.4.9 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> airtsp::SearchStringParser::definition< ScannerT >::preferred_airlines`

Definition at line 360 of file BookingRequestParser.cpp.

Referenced by airtsp::SearchStringParser::definition< ScannerT >::definition().

**10.47.4.10** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
airtsp::SearchStringParser::definition< ScannerT >::airline_code`

Definition at line 360 of file BookingRequestParser.cpp.

Referenced by airtsp::SearchStringParser::definition< ScannerT >::definition().

**10.47.4.11** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
airtsp::SearchStringParser::definition< ScannerT >::airline_name`

Definition at line 360 of file BookingRequestParser.cpp.

Referenced by airtsp::SearchStringParser::definition< ScannerT >::definition().

**10.47.4.12** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
airtsp::SearchStringParser::definition< ScannerT >::passengers`

Definition at line 360 of file BookingRequestParser.cpp.

Referenced by airtsp::SearchStringParser::definition< ScannerT >::definition().

**10.47.4.13** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
airtsp::SearchStringParser::definition< ScannerT >::passenger_number`

Definition at line 360 of file BookingRequestParser.cpp.

Referenced by airtsp::SearchStringParser::definition< ScannerT >::definition().

**10.47.4.14** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
airtsp::SearchStringParser::definition< ScannerT >::passenger_type`

Definition at line 360 of file BookingRequestParser.cpp.

Referenced by airtsp::SearchStringParser::definition< ScannerT >::definition().

**10.47.4.15** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
airtsp::SearchStringParser::definition< ScannerT >::passenger_adult_type`

Definition at line 360 of file BookingRequestParser.cpp.

Referenced by airtsp::SearchStringParser::definition< ScannerT >::definition().

**10.47.4.16** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
airtsp::SearchStringParser::definition< ScannerT >::passenger_child_type`

Definition at line 360 of file BookingRequestParser.cpp.

Referenced by airtsp::SearchStringParser::definition< ScannerT >::definition().

**10.47.4.17** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
airtsp::SearchStringParser::definition< ScannerT >::passenger_pet_type`

Definition at line 360 of file BookingRequestParser.cpp.

Referenced by airtsp::SearchStringParser::definition< ScannerT >::definition().



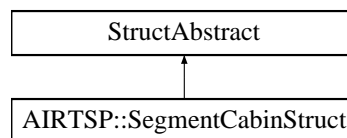
The documentation for this struct was generated from the following file:

- airtsp/batches/[BookingRequestParser.cpp](#)

## 10.48 AIRTSP::SegmentCabinStruct Struct Reference

```
#include <airtsp/bom/SegmentCabinStruct.hpp>
```

Inheritance diagram for AIRTSP::SegmentCabinStruct::



### Public Member Functions

- void [fill](#) (stdair::SegmentCabin &) const
- const std::string [describe](#) () const

### Public Attributes

- stdair::CabinCode\_T [\\_cabinCode](#)
- stdair::ClassList\_String\_T [\\_classes](#)
- stdair::FamilyCode\_T [\\_itFamilyCode](#)
- stdair::CurveKey\_T [\\_itFRAT5CurveKey](#)
- stdair::CurveKey\_T [\\_itFFDisutilityCurveKey](#)
- [FareFamilyStructList\\_T](#) [\\_fareFamilies](#)

### 10.48.1 Detailed Description

Utility Structure for the parsing of SegmentCabin details.

Definition at line 24 of file SegmentCabinStruct.hpp.

### 10.48.2 Member Function Documentation

#### 10.48.2.1 void AIRTSP::SegmentCabinStruct::fill (stdair::SegmentCabin &) const

Fill the SegmentCabin objects with the attributes of the [SegmentCabinStruct](#).

Definition at line 22 of file SegmentCabinStruct.cpp.

#### 10.48.2.2 const std::string AIRTSP::SegmentCabinStruct::describe () const

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

Definition at line 15 of file SegmentCabinStruct.cpp.

References [\\_cabinCode](#), and [\\_classes](#).

### 10.48.3 Member Data Documentation

#### 10.48.3.1 stdair::CabinCode\_T AIRTSP::SegmentCabinStruct::\_cabinCode

Definition at line 26 of file SegmentCabinStruct.hpp.

Referenced by AIRTSP::FlightPeriodStruct::addFareFamily(), describe(), and AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator()().

#### 10.48.3.2 stdair::ClassList\_String\_T AIRTSP::SegmentCabinStruct::\_classes

Definition at line 27 of file SegmentCabinStruct.hpp.

Referenced by describe(), and AIRTSP::ScheduleParserHelper::storeClasses::operator()().

#### 10.48.3.3 stdair::FamilyCode\_T AIRTSP::SegmentCabinStruct::\_itFamilyCode

Definition at line 28 of file SegmentCabinStruct.hpp.

Referenced by AIRTSP::ScheduleParserHelper::storeFClasses::operator()(), and AIRTSP::ScheduleParserHelper::storeFamilyCode::operator()().

#### 10.48.3.4 stdair::CurveKey\_T AIRTSP::SegmentCabinStruct::\_itFRAT5CurveKey

Definition at line 29 of file SegmentCabinStruct.hpp.

Referenced by AIRTSP::ScheduleParserHelper::storeFClasses::operator()(), and AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator()().

#### 10.48.3.5 stdair::CurveKey\_T AIRTSP::SegmentCabinStruct::\_itFFDisutilityCurveKey

Definition at line 30 of file SegmentCabinStruct.hpp.

Referenced by AIRTSP::ScheduleParserHelper::storeFClasses::operator()(), and AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator()().

#### 10.48.3.6 FareFamilyStructList\_T AIRTSP::SegmentCabinStruct::\_fareFamilies

Definition at line 31 of file SegmentCabinStruct.hpp.

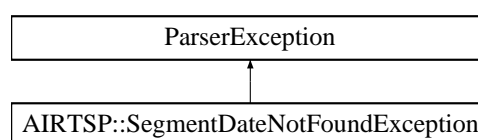
The documentation for this struct was generated from the following files:

- airtsp/bom/SegmentCabinStruct.hpp
- airtsp/bom/SegmentCabinStruct.cpp

## 10.49 AIRTSP::SegmentDateNotFoundException Class Reference

```
#include <airtsp/AIRTSP_Types.hpp>
```

Inheritance diagram for AIRTSP::SegmentDateNotFoundException:



**Public Member Functions**

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

**10.49.1 Detailed Description**

Specific exception when some BOM objects can not be found within the schedule.

Definition at line 23 of file AIRTSP\_Types.hpp.

**10.49.2 Constructor & Destructor Documentation****10.49.2.1 AIRTSP::SegmentDateNotFoundException::SegmentDateNotFoundException (const std::string &iWhat) [inline]**

Constructor.

Definition at line 28 of file AIRTSP\_Types.hpp.

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

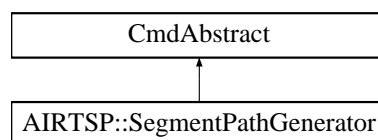
- airtsp/[AIRTSP\\_Types.hpp](#)

**10.50 AIRTSP::SegmentPathGenerator Class Reference**

Class handling the generation / instantiation of the network BOM.

```
#include <airtsp/command/SegmentPathGenerator.hpp>
```

Inheritance diagram for AIRTSP::SegmentPathGenerator::

**Static Public Member Functions**

- static void [createSegmentPathNetwork](#) (const stdair::BomRoot &)

**10.50.1 Detailed Description**

Class handling the generation / instantiation of the network BOM.

Definition at line 34 of file SegmentPathGenerator.hpp.

**10.50.2 Member Function Documentation****10.50.2.1 void AIRTSP::SegmentPathGenerator::createSegmentPathNetwork (const stdair::BomRoot &) [static]**

Generate the segment path network.

Definition at line 26 of file SegmentPathGenerator.cpp.

Referenced by AIRTSP::AIRTSP\_Service::buildComplementaryLinks().

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

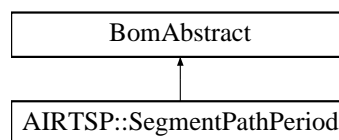
- airtsp/command/[SegmentPathGenerator.hpp](#)
- airtsp/command/[SegmentPathGenerator.cpp](#)

## 10.51 AIRTSP::SegmentPathPeriod Class Reference

Class representing a segment/path.

```
#include <airtsp/bom/SegmentPathPeriod.hpp>
```

Inheritance diagram for AIRTSP::SegmentPathPeriod::



### Public Types

- typedef [SegmentPathPeriodKey](#) Key\_T

### Public Member Functions

- const [Key\\_T](#) & [getKey](#) () const
- stdair::BomAbstract \*const [getParent](#) () const
- const stdair::PeriodStruct & [getDeparturePeriod](#) () const
- const [DateOffsetList\\_T](#) & [getBoardingDateOffsetList](#) () const
- const stdair::NbOfSegments\_T [getNbOfSegments](#) () const
- const stdair::NbOfAirlines\_T & [getNbOfAirlines](#) () const
- const stdair::Duration\_T & [getElapsedTime](#) () const
- const stdair::Duration\_T & [getBoardingTime](#) () const
- const stdair::HolderMap\_T & [getHolderMap](#) () const
- stdair::SegmentPeriod \* [getLastSegmentPeriod](#) () const
- stdair::SegmentPeriod \* [getFirstSegmentPeriod](#) () const
- const stdair::AirportCode\_T & [getDestination](#) () const
- [Key\\_T](#) [connectWithAnotherSegment](#) (const [SegmentPathPeriod](#) &) const
- bool [checkCircle](#) (const stdair::AirportCode\_T &) const
- bool [isAirlineFlown](#) (const stdair::AirlineCode\_T &) const
- bool [isDepartureDateValid](#) (const stdair::Date\_T &) const
- void [toStream](#) (std::ostream &ioOut) const
- void [fromStream](#) (std::istream &ioIn)
- std::string [toString](#) () const
- const std::string [describeKey](#) () const
- template<class Archive> void [serialize](#) (Archive &ar, const unsigned int iFileVersion)

### Protected Member Functions

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

### Protected Attributes

- [Key\\_T \\_key](#)
- [stdair::BomAbstract \\* \\_parent](#)
- [stdair::HolderMap\\_T \\_holderMap](#)

### Friends

- class [stdair::FacBom](#)
- class [stdair::FacBomManager](#)
- class [boost::serialization::access](#)

#### 10.51.1 Detailed Description

Class representing a segment/path.

It corresponds to an actual travel solution from the origin to the destination, that is, a path that a traveller can take with actual scheduled flights.

Definition at line 39 of file [SegmentPathPeriod.hpp](#).

#### 10.51.2 Member Typedef Documentation

##### 10.51.2.1 typedef [SegmentPathPeriodKey](#) AIRTSP::SegmentPathPeriod::Key\_T

Definition allowing to retrieve the associated BOM key type.

Definition at line 52 of file [SegmentPathPeriod.hpp](#).

#### 10.51.3 Constructor & Destructor Documentation

##### 10.51.3.1 AIRTSP::SegmentPathPeriod::SegmentPathPeriod (const [Key\\_T](#) &) [protected]

Main constructor.

Definition at line 43 of file [SegmentPathPeriod.cpp](#).

##### 10.51.3.2 AIRTSP::SegmentPathPeriod::~~SegmentPathPeriod () [protected]

Destructor.

Definition at line 48 of file [SegmentPathPeriod.cpp](#).

#### 10.51.4 Member Function Documentation

##### 10.51.4.1 const [Key\\_T](#)& AIRTSP::SegmentPathPeriod::getKey () const [inline]

Get the primary key (destination airport).

Definition at line 60 of file SegmentPathPeriod.hpp.

References `_key`.

**10.51.4.2** `stdair::BomAbstract* const AIRTSP::SegmentPathPeriod::getParent () const [inline]`

Get the parent (i.e., [OriginDestinationSet](#)) object.

Definition at line 67 of file SegmentPathPeriod.hpp.

References `_parent`.

**10.51.4.3** `const stdair::PeriodStruct& AIRTSP::SegmentPathPeriod::getDeparturePeriod () const [inline]`

Get the departure period (part of the primary key).

Definition at line 72 of file SegmentPathPeriod.hpp.

References `_key`, and `AIRTSP::SegmentPathPeriodKey::getPeriod()`.

Referenced by `connectWithAnotherSegment()`, and `isDepartureDateValid()`.

**10.51.4.4** `const DateOffsetList_T& AIRTSP::SegmentPathPeriod::getBoardingDateOffsetList () const [inline]`

Get the boarding date offset list (part of the primary key).

Definition at line 77 of file SegmentPathPeriod.hpp.

References `_key`, and `AIRTSP::SegmentPathPeriodKey::getBoardingDateOffsetList()`.

Referenced by `connectWithAnotherSegment()`.

**10.51.4.5** `const stdair::NbOfSegments_T AIRTSP::SegmentPathPeriod::getNbOfSegments () const [inline]`

Get the number of segments (part of the primary key).

Definition at line 82 of file SegmentPathPeriod.hpp.

References `_key`, and `AIRTSP::SegmentPathPeriodKey::getNbOfSegments()`.

Referenced by `connectWithAnotherSegment()`.

**10.51.4.6** `const stdair::NbOfAirlines_T& AIRTSP::SegmentPathPeriod::getNbOfAirlines () const [inline]`

Get the number of airlines (part of the primary key).

Definition at line 87 of file SegmentPathPeriod.hpp.

References `_key`, and `AIRTSP::SegmentPathPeriodKey::getNbOfAirlines()`.

**10.51.4.7** `const stdair::Duration_T& AIRTSP::SegmentPathPeriod::getElapsedTime () const [inline]`

Get the elapsed time (part of the primary key).

Definition at line 92 of file SegmentPathPeriod.hpp.

References `_key`, and `AIRTSP::SegmentPathPeriodKey::getElapsedTime()`.

Referenced by `connectWithAnotherSegment()`.

**10.51.4.8** `const stdair::Duration_T& AIRTSP::SegmentPathPeriod::getBoardingTime () const [inline]`

Get the boarding time (part of the primary key).

Definition at line 97 of file `SegmentPathPeriod.hpp`.

References `_key`, and `AIRTSP::SegmentPathPeriodKey::getBoardingTime()`.

Referenced by `connectWithAnotherSegment()`.

**10.51.4.9** `const stdair::HolderMap_T& AIRTSP::SegmentPathPeriod::getHolderMap () const [inline]`

Get the map of children holders (SegmentPeriod objects).

Definition at line 104 of file `SegmentPathPeriod.hpp`.

References `_holderMap`.

**10.51.4.10** `stdair::SegmentPeriod * AIRTSP::SegmentPathPeriod::getLastSegmentPeriod () const`

Get the last SegmentPeriod object of the list.

Return a NULL pointer if the list is empty.

Definition at line 91 of file `SegmentPathPeriod.cpp`.

Referenced by `connectWithAnotherSegment()`, and `getDestination()`.

**10.51.4.11** `stdair::SegmentPeriod * AIRTSP::SegmentPathPeriod::getFirstSegmentPeriod () const`

Get the first SegmentPeriod object of the list.

Return a NULL pointer if the list is empty.

Definition at line 109 of file `SegmentPathPeriod.cpp`.

Referenced by `connectWithAnotherSegment()`.

**10.51.4.12** `const stdair::AirportCode_T & AIRTSP::SegmentPathPeriod::getDestination () const`

Get the destination of the segment path (i.e., the destination of the last segment).

Definition at line 127 of file `SegmentPathPeriod.cpp`.

References `getLastSegmentPeriod()`.

**10.51.4.13** `SegmentPathPeriodKey AIRTSP::SegmentPathPeriod::connectWithAnotherSegment (const SegmentPathPeriod &) const`

Check whether the (i-1)-length segment path period can be merged with the single segment path period in order to create an i-length segment path period. The function will return a valid or non-valid segment path

period key.

The two segment path period above can be fused (and will produce a valid new segment path period key) if:

1. A passenger can connect from the last segment of the first segment path and the first segment of the next segment path. These two segments should not create another segment.
2. There is no circle within the new segment path.
3. The intersection of the two periods is non-empty.

Definition at line 163 of file SegmentPathPeriod.cpp.

References `checkCircle()`, `getBoardingDateOffsetList()`, `getBoardingTime()`, `getDeparturePeriod()`, `getElapsedTime()`, `getFirstSegmentPeriod()`, `getLastSegmentPeriod()`, `getNbOfSegments()`, `AIRTSP::SegmentPathPeriodKey::setBoardingDateOffsetList()`, `AIRTSP::SegmentPathPeriodKey::setBoardingTime()`, `AIRTSP::SegmentPathPeriodKey::setElapsedTime()`, and `AIRTSP::SegmentPathPeriodKey::setPeriod()`.

#### 10.51.4.14 **bool AIRTSP::SegmentPathPeriod::checkCircle (const stdair::AirportCode\_T &) const**

Check whether the given destination airport is also the departure point of one of the other segment members. If yes, a circle exists.

Referenced by `connectWithAnotherSegment()`.

#### 10.51.4.15 **bool AIRTSP::SegmentPathPeriod::isAirlineFlown (const stdair::AirlineCode\_T &) const**

State whether or not the given airline is flown by (at least) one of the segments of the internal list.

Definition at line 135 of file SegmentPathPeriod.cpp.

#### 10.51.4.16 **bool AIRTSP::SegmentPathPeriod::isDepartureDateValid (const stdair::Date\_T &) const**

Check whether the given departure date is included in the departure period of the segment path.

Definition at line 308 of file SegmentPathPeriod.cpp.

References `getDeparturePeriod()`.

#### 10.51.4.17 **void AIRTSP::SegmentPathPeriod::toStream (std::ostream & ioOut) const** [inline]

Dump a Business Object into an output stream.

##### Parameters:

*ostream&* the output stream.

Definition at line 176 of file SegmentPathPeriod.hpp.

References `toString()`.



**10.51.4.18 void AIRTSP::SegmentPathPeriod::fromStream (std::istream & *ioIn*) [inline]**

Read a Business Object from an input stream.

**Parameters:**

*istream&* the input stream.

Definition at line 185 of file SegmentPathPeriod.hpp.

**10.51.4.19 std::string AIRTSP::SegmentPathPeriod::toString () const**

Get the serialised version of the Business Object.

Definition at line 52 of file SegmentPathPeriod.cpp.

References `_key`, and `AIRTSP::SegmentPathPeriodKey::toString()`.

Referenced by `toStream()`.

**10.51.4.20 const std::string AIRTSP::SegmentPathPeriod::describeKey () const [inline]**

Get a string describing the key.

Definition at line 196 of file SegmentPathPeriod.hpp.

References `_key`, and `AIRTSP::SegmentPathPeriodKey::toString()`.

**10.51.4.21 template<class Archive> void AIRTSP::SegmentPathPeriod::serialize (Archive & *ar*, const unsigned int *iFileVersion*)**

Serialisation.

Definition at line 74 of file SegmentPathPeriod.cpp.

References `_key`.

**10.51.5 Friends And Related Function Documentation****10.51.5.1 friend class stdair::FacBom [friend]**

Friend classes.

Definition at line 43 of file SegmentPathPeriod.hpp.

**10.51.5.2 friend class stdair::FacBomManager [friend]**

Definition at line 44 of file SegmentPathPeriod.hpp.

**10.51.5.3 friend class boost::serialization::access [friend]**

Definition at line 45 of file SegmentPathPeriod.hpp.

**10.51.6 Member Data Documentation****10.51.6.1 [Key\\_T AIRTSP::SegmentPathPeriod::\\_key](#) [protected]**

Primary key (segment/path characteristics: scheduled period, number of segments, number of airlines, elapsed time, boarding time).

Definition at line 249 of file SegmentPathPeriod.hpp.

Referenced by describeKey(), getBoardingDateOffsetList(), getBoardingTime(), getDeparturePeriod(), getElapsedTime(), getKey(), getNbOfAirlines(), getNbOfSegments(), serialize(), and toString().

#### 10.51.6.2 stdair::BomAbstract\* AIRTSP::SegmentPathPeriod::\_parent [protected]

Pointer on the parent ([OriginDestinationSet](#)) object.

Definition at line 254 of file SegmentPathPeriod.hpp.

Referenced by getParent().

#### 10.51.6.3 stdair::HolderMap\_T AIRTSP::SegmentPathPeriod::\_holderMap [protected]

Map holding the children (SegmentPeriod objects).

#### Note:

The SegmentPeriod objects themselves have for parent the FlightPeriod class (not the [SegmentPathPeriod](#) class).

Definition at line 262 of file SegmentPathPeriod.hpp.

Referenced by getHolderMap().

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

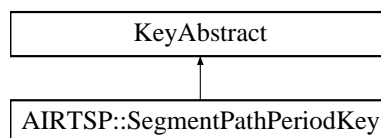
- airtsp/bom/[SegmentPathPeriod.hpp](#)
- airtsp/bom/[SegmentPathPeriod.cpp](#)

## 10.52 AIRTSP::SegmentPathPeriodKey Struct Reference

Structure representing the key of a segment/path.

```
#include <airtsp/bom/SegmentPathPeriodKey.hpp>
```

Inheritance diagram for AIRTSP::SegmentPathPeriodKey::



#### Public Member Functions

- [SegmentPathPeriodKey](#) (const stdair::PeriodStruct &, const stdair::Duration\_T &iBoardingTime, const stdair::Duration\_T &iElapsed, const [DateOffsetList\\_T](#) &, const stdair::NbOfAirlines\_T &)
- [SegmentPathPeriodKey](#) ()
- [SegmentPathPeriodKey](#) (const [SegmentPathPeriodKey](#) &)
- [~SegmentPathPeriodKey](#) ()

- const stdair::PeriodStruct & [getPeriod](#) () const
- const [DateOffsetList\\_T](#) & [getBoardingDateOffsetList](#) () const
- const stdair::NbOfSegments\_T & [getNbOfSegments](#) () const
- const stdair::NbOfAirlines\_T & [getNbOfAirlines](#) () const
- const stdair::Duration\_T & [getElapsedTime](#) () const
- const stdair::Duration\_T & [getBoardingTime](#) () const
- void [setPeriod](#) (const stdair::PeriodStruct &iPeriod)
- void [setBoardingDateOffsetList](#) (const [DateOffsetList\\_T](#) &iList)
- void [setNbOfAirlines](#) (const stdair::NbOfAirlines\_T &iNbOfAirlines)
- void [setElapsedTime](#) (const stdair::Duration\_T &iElapsed)
- void [setBoardingTime](#) (const stdair::Duration\_T &iBoardingTime)
- const bool [isValid](#) () const
- void [toStream](#) (std::ostream &ioOut) const
- void [fromStream](#) (std::istream &ioIn)
- const std::string [toString](#) () const
- template<class Archive> void [serialize](#) (Archive &ar, const unsigned int iFileVersion)

## Friends

- class [boost::serialization::access](#)

### 10.52.1 Detailed Description

Structure representing the key of a segment/path.

That key specifies a travel solution from a geographical point (origin airport) to another (destination airport).

Definition at line 33 of file SegmentPathPeriodKey.hpp.

### 10.52.2 Constructor & Destructor Documentation

**10.52.2.1 AIRTSP::SegmentPathPeriodKey::SegmentPathPeriodKey** (const stdair::PeriodStruct &, const stdair::Duration\_T & *iBoardingTime*, const stdair::Duration\_T & *iElapsed*, const [DateOffsetList\\_T](#) &, const stdair::NbOfAirlines\_T &)

Constructor.

Definition at line 40 of file SegmentPathPeriodKey.cpp.

**10.52.2.2 AIRTSP::SegmentPathPeriodKey::SegmentPathPeriodKey** ()

Default constructor.

Definition at line 22 of file SegmentPathPeriodKey.cpp.

**10.52.2.3 AIRTSP::SegmentPathPeriodKey::SegmentPathPeriodKey** (const [SegmentPathPeriodKey](#) &)

Copy constructor.

Definition at line 30 of file SegmentPathPeriodKey.cpp.

#### 10.52.2.4 AIRTSP::SegmentPathPeriodKey::~~SegmentPathPeriodKey ()

Destructor.

Definition at line 53 of file SegmentPathPeriodKey.cpp.

### 10.52.3 Member Function Documentation

#### 10.52.3.1 `const stdair::PeriodStruct& AIRTSP::SegmentPathPeriodKey::getPeriod () const` `[inline]`

Get the active days-of-week.

Definition at line 68 of file SegmentPathPeriodKey.hpp.

Referenced by AIRTSP::SegmentPathPeriod::getDeparturePeriod().

#### 10.52.3.2 `const DateOffsetList_T& AIRTSP::SegmentPathPeriodKey::getBoardingDateOffsetList () const` `[inline]`

Get the list of boarding date off-sets.

Definition at line 75 of file SegmentPathPeriodKey.hpp.

Referenced by AIRTSP::SegmentPathPeriod::getBoardingDateOffsetList().

#### 10.52.3.3 `const stdair::NbOfSegments_T AIRTSP::SegmentPathPeriodKey::getNbOfSegments () const` `[inline]`

Get the number of segments.

Definition at line 82 of file SegmentPathPeriodKey.hpp.

Referenced by AIRTSP::SegmentPathPeriod::getNbOfSegments().

#### 10.52.3.4 `const stdair::NbOfAirlines_T& AIRTSP::SegmentPathPeriodKey::getNbOfAirlines () const` `[inline]`

Get the number of airlines.

Definition at line 89 of file SegmentPathPeriodKey.hpp.

Referenced by AIRTSP::SegmentPathPeriod::getNbOfAirlines().

#### 10.52.3.5 `const stdair::Duration_T& AIRTSP::SegmentPathPeriodKey::getElapsedTime () const` `[inline]`

Get the elapsed time.

Definition at line 96 of file SegmentPathPeriodKey.hpp.

Referenced by AIRTSP::SegmentPathPeriod::getElapsedTime().

#### 10.52.3.6 `const stdair::Duration_T& AIRTSP::SegmentPathPeriodKey::getBoardingTime () const` `[inline]`

Get the boarding time.

Definition at line 103 of file SegmentPathPeriodKey.hpp.

Referenced by AIRTSP::SegmentPathPeriod::getBoardingTime().

**10.52.3.7 void AIRTSP::SegmentPathPeriodKey::setPeriod (const stdair::PeriodStruct & iPeriod)** [inline]

Set the active days-of-week.

Definition at line 111 of file SegmentPathPeriodKey.hpp.

Referenced by AIRTSP::SegmentPathPeriod::connectWithAnotherSegment().

**10.52.3.8 void AIRTSP::SegmentPathPeriodKey::setBoardingDateOffsetList (const DateOffsetList\_T & iList)** [inline]

Definition at line 115 of file SegmentPathPeriodKey.hpp.

Referenced by AIRTSP::SegmentPathPeriod::connectWithAnotherSegment().

**10.52.3.9 void AIRTSP::SegmentPathPeriodKey::setNbOfAirlines (const stdair::NbOfAirlines\_T & iNbOfAirlines)** [inline]

Set the number of airlines.

Definition at line 120 of file SegmentPathPeriodKey.hpp.

**10.52.3.10 void AIRTSP::SegmentPathPeriodKey::setElapsedTime (const stdair::Duration\_T & iElapsed)** [inline]

Set the elapsed time.

Definition at line 125 of file SegmentPathPeriodKey.hpp.

Referenced by AIRTSP::SegmentPathPeriod::connectWithAnotherSegment().

**10.52.3.11 void AIRTSP::SegmentPathPeriodKey::setBoardingTime (const stdair::Duration\_T & iBoardingTime)** [inline]

Set the boarding time.

Definition at line 130 of file SegmentPathPeriodKey.hpp.

Referenced by AIRTSP::SegmentPathPeriod::connectWithAnotherSegment().

**10.52.3.12 const bool AIRTSP::SegmentPathPeriodKey::isValid () const** [inline]

Check if the key is valid (i.e. the departure period is valid).

Definition at line 138 of file SegmentPathPeriodKey.hpp.

**10.52.3.13 void AIRTSP::SegmentPathPeriodKey::toStream (std::ostream & ioOut) const**

Dump a Business Object Key into an output stream.

#### Parameters:

*ostream&* the output stream.

Definition at line 57 of file SegmentPathPeriodKey.cpp.

References toString().

#### 10.52.3.14 void AIRTSP::SegmentPathPeriodKey::fromStream (std::istream & *ioIn*)

Read a Business Object Key from an input stream.

##### Parameters:

*istream&* the input stream.

Definition at line 62 of file SegmentPathPeriodKey.cpp.

#### 10.52.3.15 const std::string AIRTSP::SegmentPathPeriodKey::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-date.

Definition at line 66 of file SegmentPathPeriodKey.cpp.

Referenced by AIRTSP::SegmentPathPeriod::describeKey(), toStream(), and AIRTSP::SegmentPathPeriod::toString().

#### 10.52.3.16 template<class Archive> void AIRTSP::SegmentPathPeriodKey::serialize (Archive & *ar*, const unsigned int *iFileVersion*)

Serialisation.

Definition at line 98 of file SegmentPathPeriodKey.cpp.

### 10.52.4 Friends And Related Function Documentation

#### 10.52.4.1 friend class boost::serialization::access [friend]

Definition at line 34 of file SegmentPathPeriodKey.hpp.

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

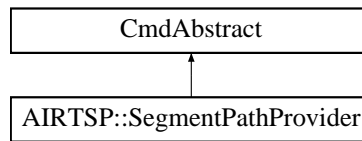
- [airtsp/bom/SegmentPathPeriodKey.hpp](#)
- [airtsp/bom/SegmentPathPeriodKey.cpp](#)

### 10.53 AIRTSP::SegmentPathProvider Class Reference

Class building the travel solutions from airline schedules.

```
#include <airtsp/command/SegmentPathProvider.hpp>
```

Inheritance diagram for AIRTSP::SegmentPathProvider::



## Friends

- class [AIRTSP\\_Service](#)

### 10.53.1 Detailed Description

Class building the travel solutions from airline schedules.

Definition at line 27 of file SegmentPathProvider.hpp.

### 10.53.2 Friends And Related Function Documentation

#### 10.53.2.1 friend class [AIRTSP\\_Service](#) [friend]

Definition at line 28 of file SegmentPathProvider.hpp.

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

- airtsp/command/[SegmentPathProvider.hpp](#)
- airtsp/command/[SegmentPathProvider.cpp](#)

## 10.54 AIRTSP::SegmentPeriodHelper Class Reference

Class representing the actual business functions for an airline segment-period.

```
#include <airtsp/bom/SegmentPeriodHelper.hpp>
```

### Static Public Member Functions

- static void [fill](#) (stdair::SegmentPeriod &, const [SegmentStruct](#) &)
- static void [fill](#) (stdair::SegmentPeriod &, const [LegStructList\\_T](#) &)

### 10.54.1 Detailed Description

Class representing the actual business functions for an airline segment-period.

Definition at line 22 of file SegmentPeriodHelper.hpp.

### 10.54.2 Member Function Documentation

#### 10.54.2.1 void AIRTSP::SegmentPeriodHelper::fill (stdair::SegmentPeriod &, const [SegmentStruct](#) &) [static]

Fill the attributes of the given segment-period with the cabins and classes.

Definition at line 14 of file SegmentPeriodHelper.cpp.

References AIRTSP::SegmentStruct::\_cabinList.

#### 10.54.2.2 void AIRTSP::SegmentPeriodHelper::fill (stdair::SegmentPeriod &, const LegStructList\_T &) [static]

Fill the attributes of the given segment-period with the list of used legs.

Definition at line 29 of file SegmentPeriodHelper.cpp.

References AIRTSP::LegStruct::\_offDateOffset, and AIRTSP::LegStruct::\_offTime.

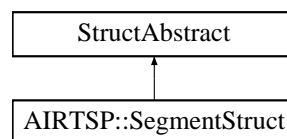
The documentation for this class was generated from the following files:

- airtsp/bom/SegmentPeriodHelper.hpp
- airtsp/bom/SegmentPeriodHelper.cpp

## 10.55 AIRTSP::SegmentStruct Struct Reference

```
#include <airtsp/bom/SegmentStruct.hpp>
```

Inheritance diagram for AIRTSP::SegmentStruct:



### Public Member Functions

- void fill (stdair::SegmentDate &) const
- const std::string describe () const

### Public Attributes

- stdair::AirportCode\_T \_boardingPoint
- stdair::Date\_T \_boardingDate
- stdair::Duration\_T \_boardingTime
- stdair::AirportCode\_T \_offPoint
- stdair::Date\_T \_offDate
- stdair::Duration\_T \_offTime
- stdair::Duration\_T \_elapsed
- SegmentCabinStructList\_T \_cabinList

#### 10.55.1 Detailed Description

Utility Structure for the parsing of Segment structures.

Definition at line 24 of file SegmentStruct.hpp.



## 10.55.2 Member Function Documentation

### 10.55.2.1 void AIRTSP::SegmentStruct::fill (stdair::SegmentDate &) const

Fill the SegmentDate objects with the attributes of the [SegmentStruct](#).

Definition at line 35 of file SegmentStruct.cpp.

### 10.55.2.2 const std::string AIRTSP::SegmentStruct::describe () const

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

Definition at line 15 of file SegmentStruct.cpp.

References `_boardingPoint`, `_boardingTime`, `_cabinList`, `_elapsed`, `_offPoint`, and `_offTime`.

## 10.55.3 Member Data Documentation

### 10.55.3.1 stdair::AirportCode\_T AIRTSP::SegmentStruct::\_boardingPoint

Definition at line 26 of file SegmentStruct.hpp.

Referenced by `AIRTSP::FlightPeriodStruct::addFareFamily()`, `AIRTSP::FlightPeriodStruct::addSegmentCabin()`, `describe()`, and `AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator()`.

### 10.55.3.2 stdair::Date\_T AIRTSP::SegmentStruct::\_boardingDate

Definition at line 27 of file SegmentStruct.hpp.

### 10.55.3.3 stdair::Duration\_T AIRTSP::SegmentStruct::\_boardingTime

Definition at line 28 of file SegmentStruct.hpp.

Referenced by `describe()`.

### 10.55.3.4 stdair::AirportCode\_T AIRTSP::SegmentStruct::\_offPoint

Definition at line 29 of file SegmentStruct.hpp.

Referenced by `AIRTSP::FlightPeriodStruct::addFareFamily()`, `AIRTSP::FlightPeriodStruct::addSegmentCabin()`, `describe()`, and `AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator()`.

### 10.55.3.5 stdair::Date\_T AIRTSP::SegmentStruct::\_offDate

Definition at line 30 of file SegmentStruct.hpp.

### 10.55.3.6 stdair::Duration\_T AIRTSP::SegmentStruct::\_offTime

Definition at line 31 of file SegmentStruct.hpp.

Referenced by `describe()`.

### 10.55.3.7 stdair::Duration\_T AIRTSP::SegmentStruct::\_elapsed

Definition at line 32 of file SegmentStruct.hpp.

Referenced by `describe()`.

### 10.55.3.8 [SegmentCabinStructList\\_T AIRTSP::SegmentStruct::\\_cabinList](#)

Definition at line 33 of file [SegmentStruct.hpp](#).

Referenced by [describe\(\)](#), and [AIRTSP::SegmentPeriodHelper::fill\(\)](#).

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

- [airtsp/bom/SegmentStruct.hpp](#)
- [airtsp/bom/SegmentStruct.cpp](#)

## 10.56 AIRTSP::ServiceAbstract Class Reference

```
#include <airtsp/service/ServiceAbstract.hpp>
```

### Public Member Functions

- virtual [~ServiceAbstract](#) ()
- virtual void [toStream](#) (std::ostream &ioOut) const
- virtual void [fromStream](#) (std::istream &ioIn)

### Protected Member Functions

- [ServiceAbstract](#) ()

### 10.56.1 Detailed Description

Base class for the Service layer.

Definition at line 14 of file [ServiceAbstract.hpp](#).

### 10.56.2 Constructor & Destructor Documentation

#### 10.56.2.1 [virtual AIRTSP::ServiceAbstract::~~ServiceAbstract](#) () [inline, virtual]

Destructor.

Definition at line 18 of file [ServiceAbstract.hpp](#).

#### 10.56.2.2 [AIRTSP::ServiceAbstract::ServiceAbstract](#) () [inline, protected]

Protected Default Constructor to ensure this class is abstract.

Definition at line 30 of file [ServiceAbstract.hpp](#).

### 10.56.3 Member Function Documentation

#### 10.56.3.1 [virtual void AIRTSP::ServiceAbstract::toStream](#) (std::ostream & ioOut) const [inline, virtual]

Dump a Business Object into an output stream.

**Parameters:**

*ostream&* the output stream.

Definition at line 22 of file ServiceAbstract.hpp.

### 10.56.3.2 virtual void AIRTSP::ServiceAbstract::fromStream (std::istream & *ioIn*) [inline, virtual]

Read a Business Object from an input stream.

**Parameters:**

*istream&* the input stream.

Definition at line 26 of file ServiceAbstract.hpp.

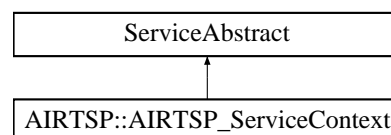
Referenced by operator>>().

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

- airtsp/service/[ServiceAbstract.hpp](#)

## 10.57 ServiceAbstract Class Reference

Inheritance diagram for ServiceAbstract::



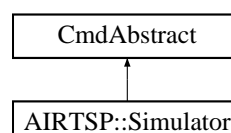
The documentation for this class was generated from the following file:

- airtsp/service/[AIRTSP\\_ServiceContext.hpp](#)

## 10.58 AIRTSP::Simulator Class Reference

```
#include <airtsp/command/Simulator.hpp>
```

Inheritance diagram for AIRTSP::Simulator::

**Static Public Member Functions**

- static void [simulate](#) (stdair::BomRoot &)

### 10.58.1 Detailed Description

Class implementing a small simulation, which uses the Airline Schedule.

Definition at line 18 of file Simulator.hpp.

### 10.58.2 Member Function Documentation

#### 10.58.2.1 void AIRTSP::Simulator::simulate (stdair::BomRoot &) [static]

Perform a small simulation, which uses the Airline Schedule.

##### Parameters:

*stdair::BomRoot&* Root of the BOM tree.

Definition at line 19 of file Simulator.cpp.

Referenced by AIRTSP::AIRTSP\_Service::simulate().

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

- airtsp/command/[Simulator.hpp](#)
- airtsp/command/[Simulator.cpp](#)

## 10.59 airtsp::store\_adult\_passenger\_type Struct Reference

### Public Member Functions

- [store\\_adult\\_passenger\\_type](#) ([SearchString\\_T](#) &ioSearchString)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [SearchString\\_T](#) & [\\_searchString](#)

### 10.59.1 Detailed Description

Store the parsed passenger type.

Definition at line 145 of file BookingRequestParser.cpp.

### 10.59.2 Constructor & Destructor Documentation

#### 10.59.2.1 airtsp::store\_adult\_passenger\_type::store\_adult\_passenger\_type ([SearchString\\_T](#) & *ioSearchString*) [inline]

Constructor.

Definition at line 147 of file BookingRequestParser.cpp.

### 10.59.3 Member Function Documentation

**10.59.3.1** `void airtsp::store_adult_passenger_type::operator() (iterator\_t iStr, iterator\_t iStrEnd) const` `[inline]`

Parse adult passenger type.

Definition at line 151 of file BookingRequestParser.cpp.

References `airtsp::SearchString_T::_passengerList`, `_searchString`, `airtsp::SearchString_T::_tmpPassenger`, `airtsp::Passenger_T::_type`, and `airtsp::Passenger_T::ADULT`.

### 10.59.4 Member Data Documentation

**10.59.4.1** `SearchString\_T & airtsp::store_adult_passenger_type::_searchString`

Definition at line 160 of file BookingRequestParser.cpp.

Referenced by `operator()()`.

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

- `airtsp/batches/BookingRequestParser.cpp`

## 10.60 airtsp::store\_airline\_code Struct Reference

### Public Member Functions

- `store_airline_code` (`SearchString\_T` &`ioSearchString`)
- `void operator() (iterator\_t iStr, iterator\_t iStrEnd) const`

### Public Attributes

- `SearchString\_T & _searchString`

### 10.60.1 Detailed Description

Store the parsed airline code.

Definition at line 92 of file BookingRequestParser.cpp.

### 10.60.2 Constructor & Destructor Documentation

**10.60.2.1** `airtsp::store_airline_code::store_airline_code (SearchString\_T & ioSearchString)` `[inline]`

Constructor.

Definition at line 94 of file BookingRequestParser.cpp.

### 10.60.3 Member Function Documentation

**10.60.3.1** `void airtsp::store_airline_code::operator() (iterator\_t iStr, iterator\_t iStrEnd) const` `[inline]`

Parse the airline code.

Definition at line 98 of file BookingRequestParser.cpp.

References airtsp::SearchString\_T::\_airlineList, airtsp::Airline\_T::\_code, \_searchString, and airtsp::SearchString\_T::\_tmpAirline.

#### 10.60.4 Member Data Documentation

##### 10.60.4.1 SearchString\_T & airtsp::store\_airline\_code::\_searchString

Definition at line 107 of file BookingRequestParser.cpp.

Referenced by operator()().

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

- airtsp/batches/BookingRequestParser.cpp

## 10.61 airtsp::store\_airline\_name Struct Reference

### Public Member Functions

- [store\\_airline\\_name](#) ([SearchString\\_T](#) &ioSearchString)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [SearchString\\_T](#) & [\\_searchString](#)

#### 10.61.1 Detailed Description

Store the parsed airline name.

Definition at line 111 of file BookingRequestParser.cpp.

#### 10.61.2 Constructor & Destructor Documentation

##### 10.61.2.1 airtsp::store\_airline\_name::store\_airline\_name ([SearchString\\_T](#) & *ioSearchString*) [\[inline\]](#)

Constructor.

Definition at line 113 of file BookingRequestParser.cpp.

#### 10.61.3 Member Function Documentation

##### 10.61.3.1 void airtsp::store\_airline\_name::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const [\[inline\]](#)

Parse the airline name.

Definition at line 117 of file BookingRequestParser.cpp.

References airtsp::SearchString\_T::\_airlineList, airtsp::Airline\_T::\_name, \_searchString, and airtsp::SearchString\_T::\_tmpAirline.

### 10.61.4 Member Data Documentation

#### 10.61.4.1 [SearchString\\_T](#) & [airtsp::store\\_airline\\_name::\\_searchString](#)

Definition at line 126 of file `BookingRequestParser.cpp`.

Referenced by `operator()()`.

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

- `airtsp/batches/BookingRequestParser.cpp`

## 10.62 airtsp::store\_airline\_sign Struct Reference

### Public Member Functions

- [store\\_airline\\_sign](#) ([SearchString\\_T](#) & *ioSearchString*)
- [operator\(\)](#) (bool *iAirlineSign*) const

### Public Attributes

- [SearchString\\_T](#) & [\\_searchString](#)

### 10.62.1 Detailed Description

Store the airline sign (+/-).

Definition at line 77 of file `BookingRequestParser.cpp`.

### 10.62.2 Constructor & Destructor Documentation

#### 10.62.2.1 [airtsp::store\\_airline\\_sign::store\\_airline\\_sign](#) ([SearchString\\_T](#) & *ioSearchString*) [inline]

Constructor.

Definition at line 79 of file `BookingRequestParser.cpp`.

### 10.62.3 Member Function Documentation

#### 10.62.3.1 [void airtsp::store\\_airline\\_sign::operator\(\)](#) (bool *iAirlineSign*) const [inline]

Parse the airline sign.

Definition at line 83 of file `BookingRequestParser.cpp`.

References `airtsp::Airline_T::_isPreferred`, `_searchString`, and `airtsp::SearchString_T::_tmpAirline`.

### 10.62.4 Member Data Documentation

#### 10.62.4.1 [SearchString\\_T](#) & [airtsp::store\\_airline\\_sign::\\_searchString](#)

Definition at line 88 of file `BookingRequestParser.cpp`.

Referenced by `operator()()`.

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

- [airtsp/batches/BookingRequestParser.cpp](#)

## 10.63 airtsp::store\_child\_passenger\_type Struct Reference

### Public Member Functions

- [store\\_child\\_passenger\\_type](#) ([SearchString\\_T](#) &ioSearchString)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [SearchString\\_T](#) & [\\_searchString](#)

#### 10.63.1 Detailed Description

Store the parsed passenger type.

Definition at line 164 of file [BookingRequestParser.cpp](#).

#### 10.63.2 Constructor & Destructor Documentation

##### 10.63.2.1 airtsp::store\_child\_passenger\_type::store\_child\_passenger\_type ([SearchString\\_T](#) & *ioSearchString*) [\[inline\]](#)

Constructor.

Definition at line 166 of file [BookingRequestParser.cpp](#).

#### 10.63.3 Member Function Documentation

##### 10.63.3.1 void airtsp::store\_child\_passenger\_type::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const [\[inline\]](#)

Parse child passenger type.

Definition at line 170 of file [BookingRequestParser.cpp](#).

References [airtsp::SearchString\\_T::\\_passengerList](#), [\\_searchString](#), [airtsp::SearchString\\_T::\\_tmpPassenger](#), [airtsp::Passenger\\_T::\\_type](#), and [airtsp::Passenger\\_T::CHILD](#).

#### 10.63.4 Member Data Documentation

##### 10.63.4.1 [SearchString\\_T](#) & airtsp::store\_child\_passenger\_type::\_searchString

Definition at line 179 of file [BookingRequestParser.cpp](#).

Referenced by [operator\(\)](#).

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

- [airtsp/batches/BookingRequestParser.cpp](#)



## 10.64 airtsp::store\_date Struct Reference

### Public Member Functions

- [store\\_date](#) ([SearchString\\_T](#) &ioSearchString)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [SearchString\\_T](#) & [\\_searchString](#)

#### 10.64.1 Detailed Description

Store a parsed date.

Definition at line 58 of file BookingRequestParser.cpp.

#### 10.64.2 Constructor & Destructor Documentation

##### 10.64.2.1 airtsp::store\_date::store\_date ([SearchString\\_T](#) &ioSearchString) [inline]

Constructor.

Definition at line 60 of file BookingRequestParser.cpp.

#### 10.64.3 Member Function Documentation

##### 10.64.3.1 void airtsp::store\_date::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const [inline]

Parse the date.

Definition at line 64 of file BookingRequestParser.cpp.

References [airtsp::Date\\_T::\\_date](#), [airtsp::SearchString\\_T::\\_dateList](#), [\\_searchString](#), [airtsp::SearchString\\_T::\\_tmpDate](#), and [airtsp::Date\\_T::getDate\(\)](#).

#### 10.64.4 Member Data Documentation

##### 10.64.4.1 [SearchString\\_T](#) & [airtsp::store\\_date::\\_searchString](#)

Definition at line 73 of file BookingRequestParser.cpp.

Referenced by [operator\(\)](#).

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

- [airtsp/batches/BookingRequestParser.cpp](#)

## 10.65 airtsp::store\_passenger\_number Struct Reference

### Public Member Functions

- [store\\_passenger\\_number](#) ([SearchString\\_T](#) &ioSearchString)
- void [operator\(\)](#) (unsigned int iNumber) const

### Public Attributes

- [SearchString\\_T](#) & [\\_searchString](#)

#### 10.65.1 Detailed Description

Store the parsed number of passengers.

Definition at line 130 of file BookingRequestParser.cpp.

#### 10.65.2 Constructor & Destructor Documentation

##### 10.65.2.1 airtsp::store\_passenger\_number::store\_passenger\_number ([SearchString\\_T](#) & *io-SearchString*) [[inline](#)]

Constructor.

Definition at line 132 of file BookingRequestParser.cpp.

#### 10.65.3 Member Function Documentation

##### 10.65.3.1 void airtsp::store\_passenger\_number::operator() (unsigned int *iNumber*) const [[inline](#)]

Parse number of passengers.

Definition at line 136 of file BookingRequestParser.cpp.

References [airtsp::Passenger\\_T::\\_number](#), [\\_searchString](#), and [airtsp::SearchString\\_T::\\_tmpPassenger](#).

#### 10.65.4 Member Data Documentation

##### 10.65.4.1 [SearchString\\_T](#) & airtsp::store\_passenger\_number::\_searchString

Definition at line 141 of file BookingRequestParser.cpp.

Referenced by [operator\(\)\(\)](#).

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

- [airtsp/batches/BookingRequestParser.cpp](#)

## 10.66 airtsp::store\_pet\_passenger\_type Struct Reference

### Public Member Functions

- [store\\_pet\\_passenger\\_type](#) ([SearchString\\_T](#) & [ioSearchString](#))
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [SearchString\\_T](#) & [\\_searchString](#)

### 10.66.1 Detailed Description

Store the parsed passenger type.

Definition at line 183 of file BookingRequestParser.cpp.

### 10.66.2 Constructor & Destructor Documentation

**10.66.2.1 airtsp::store\_pet\_passenger\_type::store\_pet\_passenger\_type ([SearchString\\_T](#) & *io-SearchString*) [inline]**

Constructor.

Definition at line 185 of file BookingRequestParser.cpp.

### 10.66.3 Member Function Documentation

**10.66.3.1 void airtsp::store\_pet\_passenger\_type::operator() ([iterator\\_t](#) *iStr*, [iterator\\_t](#) *iStrEnd*) const [inline]**

Parse pet passenger type.

Definition at line 189 of file BookingRequestParser.cpp.

References `airtsp::SearchString_T::_passengerList`, `_searchString`, `airtsp::SearchString_T::_tmpPassenger`, `airtsp::Passenger_T::_type`, and `airtsp::Passenger_T::PET`.

### 10.66.4 Member Data Documentation

**10.66.4.1 [SearchString\\_T](#) & airtsp::store\_pet\_passenger\_type::\_searchString**

Definition at line 198 of file BookingRequestParser.cpp.

Referenced by `operator()`.

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

- [airtsp/batches/BookingRequestParser.cpp](#)

## 10.67 airtsp::store\_place\_element Struct Reference

### Public Member Functions

- [store\\_place\\_element](#) ([SearchString\\_T](#) & *ioSearchString*)
- void [operator\(\)](#) ([iterator\\_t](#) *iStr*, [iterator\\_t](#) *iStrEnd*) const

### Public Attributes

- [SearchString\\_T](#) & [\\_searchString](#)

### 10.67.1 Detailed Description

Store the parsed place element.

Definition at line 37 of file BookingRequestParser.cpp.

### 10.67.2 Constructor & Destructor Documentation

#### 10.67.2.1 airtsp::store\_place\_element::store\_place\_element ([SearchString\\_T](#) & *ioSearchString*) [inline]

Constructor.

Definition at line 39 of file BookingRequestParser.cpp.

### 10.67.3 Member Function Documentation

#### 10.67.3.1 void airtsp::store\_place\_element::operator() ([iterator\\_t](#) *iStr*, [iterator\\_t](#) *iStrEnd*) const [inline]

Parse the place.

Definition at line 43 of file BookingRequestParser.cpp.

References [airtsp::Place\\_T::\\_name](#), [\\_searchString](#), and [airtsp::SearchString\\_T::\\_tmpPlace](#).

### 10.67.4 Member Data Documentation

#### 10.67.4.1 [SearchString\\_T](#) & airtsp::store\_place\_element::\_searchString

Definition at line 54 of file BookingRequestParser.cpp.

Referenced by [operator\(\)](#).

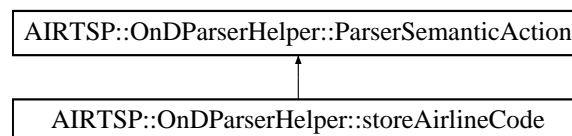
The documentation for this struct was generated from the following file:

- [airtsp/batches/BookingRequestParser.cpp](#)

## 10.68 AIRTSP::OnDParserHelper::storeAirlineCode Struct Reference

```
#include <airtsp/command/OnDParserHelper.hpp>
```

Inheritance diagram for AIRTSP::OnDParserHelper::storeAirlineCode::



### Public Member Functions

- [storeAirlineCode](#) ([OnDPeriodStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) *iStr*, [iterator\\_t](#) *iStrEnd*) const

### Public Attributes

- [OnDPeriodStruct](#) & [\\_onDPeriod](#)

### 10.68.1 Detailed Description

Store the parsed airline code.

Definition at line 90 of file OnDParserHelper.hpp.

### 10.68.2 Constructor & Destructor Documentation

#### 10.68.2.1 AIRTSP::OnDParserHelper::storeAirlineCode::storeAirlineCode ([OnDPeriodStruct](#) &)

Actor Constructor.

Definition at line 139 of file OnDParserHelper.cpp.

### 10.68.3 Member Function Documentation

#### 10.68.3.1 void AIRTSP::OnDParserHelper::storeAirlineCode::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 144 of file OnDParserHelper.cpp.

References AIRTSP::OnDPeriodStruct::\_airlineCode, AIRTSP::OnDPeriodStruct::\_airlineCodeList, AIRTSP::OnDPeriodStruct::\_nbOfAirlines, and AIRTSP::OnDParserHelper::ParserSemanticAction::\_onDPeriod.

### 10.68.4 Member Data Documentation

#### 10.68.4.1 [OnDPeriodStruct&](#) AIRTSP::OnDParserHelper::ParserSemanticAction::\_onDPeriod [inherited]

Actor Context.

Definition at line 38 of file OnDParserHelper.hpp.

Referenced by AIRTSP::OnDParserHelper::doEndOnD::operator(), AIRTSP::OnDParserHelper::storeClassCode::operator(), operator(), AIRTSP::OnDParserHelper::storeEndRangeTime::operator(), AIRTSP::OnDParserHelper::storeStartRangeTime::operator(), AIRTSP::OnDParserHelper::storeDateRangeEnd::operator(), AIRTSP::OnDParserHelper::storeDateRangeStart::operator(), AIRTSP::OnDParserHelper::storeDestination::operator(), and AIRTSP::OnDParserHelper::storeOrigin::operator().

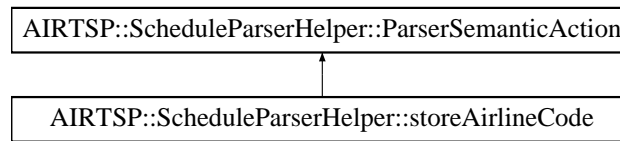
The documentation for this struct was generated from the following files:

- [airtsp/command/OnDParserHelper.hpp](#)
- [airtsp/command/OnDParserHelper.cpp](#)

## 10.69 AIRTSP::ScheduleParserHelper::storeAirlineCode Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeAirlineCode::



## Public Member Functions

- [storeAirlineCode](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

## Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

### 10.69.1 Detailed Description

Store the parsed airline code.

Definition at line 37 of file `ScheduleParserHelper.hpp`.

### 10.69.2 Constructor & Destructor Documentation

#### 10.69.2.1 AIRTSP::ScheduleParserHelper::storeAirlineCode::storeAirlineCode ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 32 of file `ScheduleParserHelper.cpp`.

### 10.69.3 Member Function Documentation

#### 10.69.3.1 void AIRTSP::ScheduleParserHelper::storeAirlineCode::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 37 of file `ScheduleParserHelper.cpp`.

References `AIRTSP::FlightPeriodStruct::_airlineCode`, `AIRTSP::ScheduleParserHelper::ParserSemanticAction::_flightPeriod`, and `AIRTSP::FlightPeriodStruct::_legList`.

### 10.69.4 Member Data Documentation

#### 10.69.4.1 [FlightPeriodStruct](#)& [AIRTSP::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#) [inherited]

Actor Context.

Definition at line 33 of file `ScheduleParserHelper.hpp`.

Referenced by `AIRTSP::ScheduleParserHelper::doEndFlight::operator()`, `AIRTSP::ScheduleParserHelper::storeFClasses::operator()`, `AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator()`, and `AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator()`.

AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and operator().

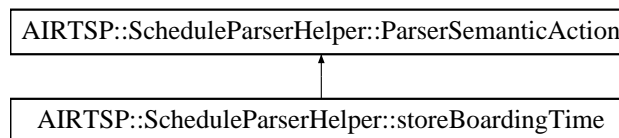
The documentation for this struct was generated from the following files:

- airtsp/command/[ScheduleParserHelper.hpp](#)
- airtsp/command/[ScheduleParserHelper.cpp](#)

## 10.70 AIRTSP::ScheduleParserHelper::storeBoardingTime Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeBoardingTime::



### Public Member Functions

- [storeBoardingTime](#) ([FlightPeriodStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.70.1 Detailed Description

Store the boarding time.

Definition at line 109 of file [ScheduleParserHelper.hpp](#).

#### 10.70.2 Constructor & Destructor Documentation

##### 10.70.2.1 AIRTSP::ScheduleParserHelper::storeBoardingTime::storeBoardingTime ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 191 of file ScheduleParserHelper.cpp.

### 10.70.3 Member Function Documentation

#### 10.70.3.1 void AIRTSP::ScheduleParserHelper::storeBoardingTime::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 196 of file ScheduleParserHelper.cpp.

References AIRTSP::LegStruct::\_boardingTime, AIRTSP::FlightPeriodStruct::\_dateOffset, AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, AIRTSP::FlightPeriodStruct::\_itLeg, AIRTSP::FlightPeriodStruct::\_itSeconds, and AIRTSP::FlightPeriodStruct::getTime().

### 10.70.4 Member Data Documentation

#### 10.70.4.1 FlightPeriodStruct& AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::doEndFlight::operator(), AIRTSP::ScheduleParserHelper::storeFCClasses::operator(), AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

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

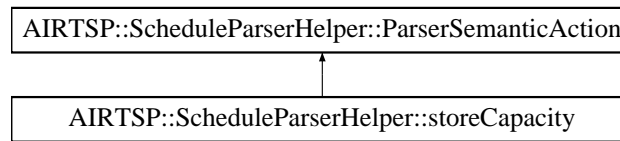
- airtsp/command/ScheduleParserHelper.hpp
- airtsp/command/ScheduleParserHelper.cpp

## 10.71 AIRTSP::ScheduleParserHelper::storeCapacity Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeCapacity::





### Public Member Functions

- [storeCapacity](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) (double *iReal*) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.71.1 Detailed Description

Store the parsed capacity.

Definition at line 141 of file `ScheduleParserHelper.hpp`.

#### 10.71.2 Constructor & Destructor Documentation

##### 10.71.2.1 AIRTSP::ScheduleParserHelper::storeCapacity::storeCapacity ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 263 of file `ScheduleParserHelper.cpp`.

#### 10.71.3 Member Function Documentation

##### 10.71.3.1 void AIRTSP::ScheduleParserHelper::storeCapacity::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 268 of file `ScheduleParserHelper.cpp`.

References `AIRTSP::LegStruct::_cabinList`, `AIRTSP::LegCabinStruct::_capacity`, `AIRTSP::ScheduleParserHelper::ParserSemanticAction::_flightPeriod`, `AIRTSP::FlightPeriodStruct::_itLeg`, and `AIRTSP::FlightPeriodStruct::_itLegCabin`.

#### 10.71.4 Member Data Documentation

##### 10.71.4.1 [FlightPeriodStruct&](#) [AIRTSP::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#) [inherited]

Actor Context.

Definition at line 33 of file `ScheduleParserHelper.hpp`.

Referenced by `AIRTSP::ScheduleParserHelper::doEndFlight::operator()`, `AIRTSP::ScheduleParserHelper::storeFClasses::operator()`, `AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator()`, and `AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator()`.

AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

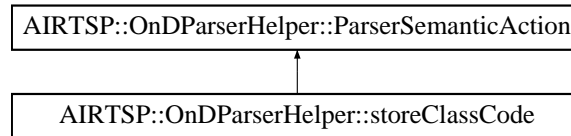
The documentation for this struct was generated from the following files:

- airtsp/command/[ScheduleParserHelper.hpp](#)
- airtsp/command/[ScheduleParserHelper.cpp](#)

## 10.72 AIRTSP::OnDParserHelper::storeClassCode Struct Reference

```
#include <airtsp/command/OnDParserHelper.hpp>
```

Inheritance diagram for AIRTSP::OnDParserHelper::storeClassCode::



### Public Member Functions

- [storeClassCode](#) ([OnDPeriodStruct](#) &)
- void [operator\(\)](#) (char iChar) const

### Public Attributes

- [OnDPeriodStruct](#) & [\\_onDPeriod](#)

#### 10.72.1 Detailed Description

Store the parsed class code.

Definition at line 98 of file OnDParserHelper.hpp.

#### 10.72.2 Constructor & Destructor Documentation

##### 10.72.2.1 AIRTSP::OnDParserHelper::storeClassCode::storeClassCode ([OnDPeriodStruct](#) &)

Actor Constructor.

Definition at line 172 of file OnDParserHelper.cpp.

### 10.72.3 Member Function Documentation

#### 10.72.3.1 void AIRTSP::OnDParserHelper::storeClassCode::operator() (char *iChar*) const

Actor Function (functor).

Definition at line 177 of file OnDParserHelper.cpp.

References AIRTSP::OnDPeriodStruct::\_classCode, AIRTSP::OnDPeriodStruct::\_classCodeList, and AIRTSP::OnDParserHelper::ParserSemanticAction::\_onDPeriod.

### 10.72.4 Member Data Documentation

#### 10.72.4.1 OnDPeriodStruct& AIRTSP::OnDParserHelper::ParserSemanticAction::\_onDPeriod [inherited]

Actor Context.

Definition at line 38 of file OnDParserHelper.hpp.

Referenced by AIRTSP::OnDParserHelper::doEndOnD::operator()(), operator()(), AIRTSP::OnDParserHelper::storeAirlineCode::operator()(), AIRTSP::OnDParserHelper::storeEndRangeTime::operator()(), AIRTSP::OnDParserHelper::storeStartRangeTime::operator()(), AIRTSP::OnDParserHelper::storeDateRangeEnd::operator()(), AIRTSP::OnDParserHelper::storeDateRangeStart::operator()(), AIRTSP::OnDParserHelper::storeDestination::operator()(), and AIRTSP::OnDParserHelper::storeOrigin::operator()().

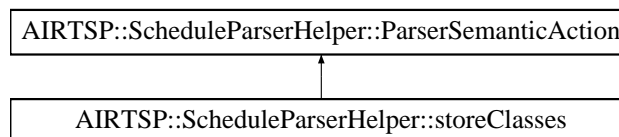
The documentation for this struct was generated from the following files:

- airtsp/command/OnDParserHelper.hpp
- airtsp/command/OnDParserHelper.cpp

## 10.73 AIRTSP::ScheduleParserHelper::storeClasses Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeClasses::



### Public Member Functions

- storeClasses (FlightPeriodStruct &)
- void operator() (iterator\_t iStr, iterator\_t iStrEnd) const

### Public Attributes

- FlightPeriodStruct & \_flightPeriod

### 10.73.1 Detailed Description

Store the parsed list of class codes.

Definition at line 184 of file ScheduleParserHelper.hpp.

### 10.73.2 Constructor & Destructor Documentation

#### 10.73.2.1 AIRTSP::ScheduleParserHelper::storeClasses::storeClasses ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 345 of file ScheduleParserHelper.cpp.

### 10.73.3 Member Function Documentation

#### 10.73.3.1 void AIRTSP::ScheduleParserHelper::storeClasses::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 350 of file ScheduleParserHelper.cpp.

References AIRTSP::FlightPeriodStruct::\_areSegmentDefinitionsSpecific, AIRTSP::SegmentCabinStruct::\_classes, AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, AIRTSP::FlightPeriodStruct::\_itSegment, AIRTSP::FlightPeriodStruct::\_itSegmentCabin, and AIRTSP::FlightPeriodStruct::addSegmentCabin().

### 10.73.4 Member Data Documentation

#### 10.73.4.1 [FlightPeriodStruct&](#) AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [*inherited*]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::doEndFlight::operator(), AIRTSP::ScheduleParserHelper::storeFCClasses::operator(), AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

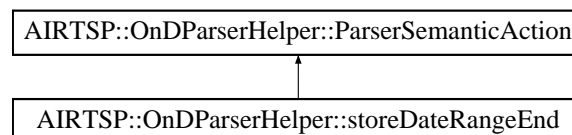
The documentation for this struct was generated from the following files:

- [airtsp/command/ScheduleParserHelper.hpp](#)
- [airtsp/command/ScheduleParserHelper.cpp](#)

## 10.74 AIRTSP::OnDParserHelper::storeDateRangeEnd Struct Reference

```
#include <airtsp/command/OnDParserHelper.hpp>
```

Inheritance diagram for AIRTSP::OnDParserHelper::storeDateRangeEnd::



### Public Member Functions

- [storeDateRangeEnd](#) ([OnDPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [OnDPeriodStruct](#) & [\\_onDPeriod](#)

#### 10.74.1 Detailed Description

Store the end of the date range.

Definition at line 66 of file OnDParserHelper.hpp.

#### 10.74.2 Constructor & Destructor Documentation

##### 10.74.2.1 AIRTSP::OnDParserHelper::storeDateRangeEnd::storeDateRangeEnd ([OnDPeriodStruct](#) &)

Actor Constructor.

Definition at line 83 of file OnDParserHelper.cpp.

#### 10.74.3 Member Function Documentation

##### 10.74.3.1 void AIRTSP::OnDParserHelper::storeDateRangeEnd::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 88 of file OnDParserHelper.cpp.

References [AIRTSP::OnDPeriodStruct::\\_datePeriod](#), [AIRTSP::OnDPeriodStruct::\\_dateRangeEnd](#), [AIRTSP::OnDPeriodStruct::\\_dateRangeStart](#), [AIRTSP::OnDPeriodStruct::\\_itSeconds](#), [AIRTSP::OnDParserHelper::ParserSemanticAction::\\_onDPeriod](#), and [AIRTSP::OnDPeriodStruct::getDate\(\)](#).

#### 10.74.4 Member Data Documentation

##### 10.74.4.1 OnDPeriodStruct& AIRTSP::OnDParserHelper::ParserSemanticAction::\_onDPeriod [inherited]

Actor Context.

Definition at line 38 of file OnDParserHelper.hpp.

Referenced by AIRTSP::OnDParserHelper::doEndOnD::operator(), AIRTSP::OnDParserHelper::storeClassCode::operator(), AIRTSP::OnDParserHelper::storeAirlineCode::operator(), AIRTSP::OnDParserHelper::storeEndRangeTime::operator(), AIRTSP::OnDParserHelper::storeStartRangeTime::operator(), operator(), AIRTSP::OnDParserHelper::storeDateRangeStart::operator(), AIRTSP::OnDParserHelper::storeDestination::operator(), and AIRTSP::OnDParserHelper::storeOrigin::operator().

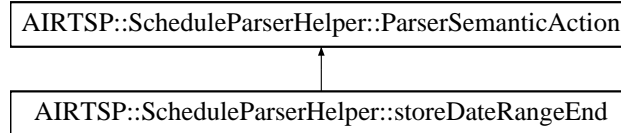
The documentation for this struct was generated from the following files:

- airtsp/command/OnDParserHelper.hpp
- airtsp/command/OnDParserHelper.cpp

## 10.75 AIRTSP::ScheduleParserHelper::storeDateRangeEnd Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeDateRangeEnd::



### Public Member Functions

- storeDateRangeEnd (FlightPeriodStruct &)
- void operator() (iterator\_t iStr, iterator\_t iStrEnd) const

### Public Attributes

- FlightPeriodStruct & \_flightPeriod

#### 10.75.1 Detailed Description

Store the end of the date range.

Definition at line 61 of file ScheduleParserHelper.hpp.

#### 10.75.2 Constructor & Destructor Documentation

##### 10.75.2.1 AIRTSP::ScheduleParserHelper::storeDateRangeEnd::storeDateRangeEnd (FlightPeriodStruct &)

Actor Constructor.

Definition at line 77 of file ScheduleParserHelper.cpp.

### 10.75.3 Member Function Documentation

#### 10.75.3.1 void AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 82 of file ScheduleParserHelper.cpp.

References AIRTSP::FlightPeriodStruct::\_airlineCode, AIRTSP::LegStruct::\_airlineCode, AIRTSP::FlightPeriodStruct::\_dateRange, AIRTSP::FlightPeriodStruct::\_dateRangeEnd, AIRTSP::FlightPeriodStruct::\_dateRangeStart, AIRTSP::FlightPeriodStruct::\_flightNumber, AIRTSP::LegStruct::\_flightNumber, AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, AIRTSP::FlightPeriodStruct::\_itLeg, AIRTSP::FlightPeriodStruct::\_itSeconds, and AIRTSP::FlightPeriodStruct::getDate().

### 10.75.4 Member Data Documentation

#### 10.75.4.1 FlightPeriodStruct& AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::doEndFlight::operator(), AIRTSP::ScheduleParserHelper::storeFClasses::operator(), AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

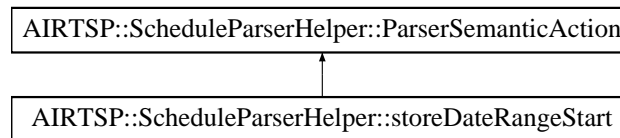
The documentation for this struct was generated from the following files:

- airtsp/command/ScheduleParserHelper.hpp
- airtsp/command/ScheduleParserHelper.cpp

## 10.76 AIRTSP::ScheduleParserHelper::storeDateRangeStart Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeDateRangeStart::



## Public Member Functions

- [storeDateRangeStart](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

## Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

### 10.76.1 Detailed Description

Store the start of the date range.

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

### 10.76.2 Constructor & Destructor Documentation

#### 10.76.2.1 AIRTSP::ScheduleParserHelper::storeDateRangeStart::storeDateRangeStart ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 62 of file `ScheduleParserHelper.cpp`.

### 10.76.3 Member Function Documentation

#### 10.76.3.1 void AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 67 of file `ScheduleParserHelper.cpp`.

References `AIRTSP::FlightPeriodStruct::_dateRangeStart`, `AIRTSP::ScheduleParserHelper::ParserSemanticAction::_flightPeriod`, `AIRTSP::FlightPeriodStruct::_itSeconds`, and `AIRTSP::FlightPeriodStruct::getDate()`.

### 10.76.4 Member Data Documentation

#### 10.76.4.1 [FlightPeriodStruct&](#) [AIRTSP::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#) [inherited]

Actor Context.

Definition at line 33 of file `ScheduleParserHelper.hpp`.



Referenced by AIRTSP::ScheduleParserHelper::doEndFlight::operator(), AIRTSP::ScheduleParserHelper::storeFCClasses::operator(), AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

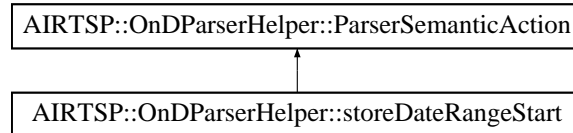
The documentation for this struct was generated from the following files:

- airtsp/command/[ScheduleParserHelper.hpp](#)
- airtsp/command/[ScheduleParserHelper.cpp](#)

## 10.77 AIRTSP::OnDParserHelper::storeDateRangeStart Struct Reference

```
#include <airtsp/command/OnDParserHelper.hpp>
```

Inheritance diagram for AIRTSP::OnDParserHelper::storeDateRangeStart::



### Public Member Functions

- [storeDateRangeStart](#) (OnDPeriodStruct &)
- void [operator\(\)](#) (iterator\_t iStr, iterator\_t iStrEnd) const

### Public Attributes

- [OnDPeriodStruct](#) & [\\_onDPeriod](#)

#### 10.77.1 Detailed Description

Store the start of the date range.

Definition at line 58 of file OnDParserHelper.hpp.

### 10.77.2 Constructor & Destructor Documentation

#### 10.77.2.1 AIRTSP::OnDParserHelper::storeDateRangeStart::storeDateRangeStart ([OnDPeriodStruct](#) &)

Actor Constructor.

Definition at line 66 of file OnDParserHelper.cpp.

### 10.77.3 Member Function Documentation

#### 10.77.3.1 void AIRTSP::OnDParserHelper::storeDateRangeStart::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 71 of file OnDParserHelper.cpp.

References AIRTSP::OnDPeriodStruct::\_dateRangeStart, AIRTSP::OnDPeriodStruct::\_itSeconds, AIRTSP::OnDParserHelper::ParserSemanticAction::\_onDPeriod, and AIRTSP::OnDPeriodStruct::getDate().

### 10.77.4 Member Data Documentation

#### 10.77.4.1 [OnDPeriodStruct&](#) AIRTSP::OnDParserHelper::ParserSemanticAction::\_onDPeriod [inherited]

Actor Context.

Definition at line 38 of file OnDParserHelper.hpp.

Referenced by AIRTSP::OnDParserHelper::doEndOnD::operator()(), AIRTSP::OnDParserHelper::storeClassCode::operator()(), AIRTSP::OnDParserHelper::storeAirlineCode::operator()(), AIRTSP::OnDParserHelper::storeEndRangeTime::operator()(), AIRTSP::OnDParserHelper::storeStartRangeTime::operator()(), AIRTSP::OnDParserHelper::storeDateRangeEnd::operator()(), operator()(), AIRTSP::OnDParserHelper::storeDestination::operator()(), and AIRTSP::OnDParserHelper::storeOrigin::operator()().

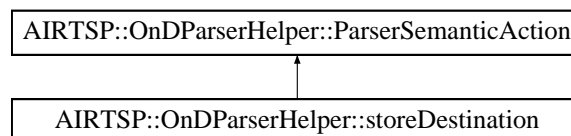
The documentation for this struct was generated from the following files:

- airtsp/command/[OnDParserHelper.hpp](#)
- airtsp/command/[OnDParserHelper.cpp](#)

## 10.78 AIRTSP::OnDParserHelper::storeDestination Struct Reference

```
#include <airtsp/command/OnDParserHelper.hpp>
```

Inheritance diagram for AIRTSP::OnDParserHelper::storeDestination::



**Public Member Functions**

- [storeDestination](#) ([OnDPeriodStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

**Public Attributes**

- [OnDPeriodStruct](#) & [\\_onDPeriod](#)

**10.78.1 Detailed Description**

Store the parsed destination.

Definition at line 50 of file OnDParserHelper.hpp.

**10.78.2 Constructor & Destructor Documentation****10.78.2.1 AIRTSP::OnDParserHelper::storeDestination::storeDestination ([OnDPeriodStruct](#) &)**

Actor Constructor.

Definition at line 50 of file OnDParserHelper.cpp.

**10.78.3 Member Function Documentation****10.78.3.1 void AIRTSP::OnDParserHelper::storeDestination::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const**

Actor Function (functor).

Definition at line 55 of file OnDParserHelper.cpp.

References [AIRTSP::OnDPeriodStruct::\\_destination](#), and [AIRTSP::OnDParserHelper::ParserSemanticAction::\\_onDPeriod](#).

**10.78.4 Member Data Documentation****10.78.4.1 [OnDPeriodStruct&](#) [AIRTSP::OnDParserHelper::ParserSemanticAction::\\_onDPeriod](#) [inherited]**

Actor Context.

Definition at line 38 of file OnDParserHelper.hpp.

Referenced by [AIRTSP::OnDParserHelper::doEndOnD::operator\(\)](#), [AIRTSP::OnDParserHelper::storeClassCode::operator\(\)](#), [AIRTSP::OnDParserHelper::storeAirlineCode::operator\(\)](#), [AIRTSP::OnDParserHelper::storeEndRangeTime::operator\(\)](#), [AIRTSP::OnDParserHelper::storeStartRangeTime::operator\(\)](#), [AIRTSP::OnDParserHelper::storeDateRangeEnd::operator\(\)](#), [AIRTSP::OnDParserHelper::storeDateRangeStart::operator\(\)](#), [operator\(\)](#), and [AIRTSP::OnDParserHelper::storeOrigin::operator\(\)](#).

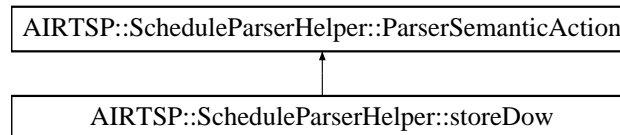
The documentation for this struct was generated from the following files:

- [airtsp/command/OnDParserHelper.hpp](#)
- [airtsp/command/OnDParserHelper.cpp](#)

## 10.79 AIRTSP::ScheduleParserHelper::storeDow Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeDow::



### Public Member Functions

- [storeDow](#) ([FlightPeriodStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.79.1 Detailed Description

Store the DOW (day of the Week).

Definition at line 69 of file ScheduleParserHelper.hpp.

#### 10.79.2 Constructor & Destructor Documentation

##### 10.79.2.1 AIRTSP::ScheduleParserHelper::storeDow::storeDow ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 104 of file ScheduleParserHelper.cpp.

#### 10.79.3 Member Function Documentation

##### 10.79.3.1 void AIRTSP::ScheduleParserHelper::storeDow::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 109 of file ScheduleParserHelper.cpp.

References [AIRTSP::FlightPeriodStruct::\\_dow](#), and [AIRTSP::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#).

#### 10.79.4 Member Data Documentation

##### 10.79.4.1 [FlightPeriodStruct](#)& [AIRTSP::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#) [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::doEndFlight::operator(), AIRTSP::ScheduleParserHelper::storeFCClasses::operator(), AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

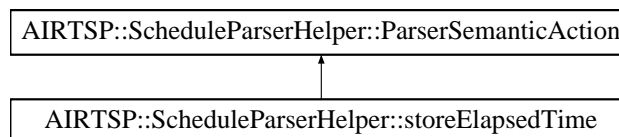
The documentation for this struct was generated from the following files:

- airtsp/command/ScheduleParserHelper.hpp
- airtsp/command/ScheduleParserHelper.cpp

## 10.80 AIRTSP::ScheduleParserHelper::storeElapsedTime Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeElapsedTime::



### Public Member Functions

- [storeElapsedTime](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.80.1 Detailed Description

Store the elapsed time.

Definition at line 125 of file ScheduleParserHelper.hpp.

## 10.80.2 Constructor & Destructor Documentation

### 10.80.2.1 AIRTSP::ScheduleParserHelper::storeElapsedTime::storeElapsedTime ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 230 of file ScheduleParserHelper.cpp.

## 10.80.3 Member Function Documentation

### 10.80.3.1 void AIRTSP::ScheduleParserHelper::storeElapsedTime::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 235 of file ScheduleParserHelper.cpp.

References AIRTSP::FlightPeriodStruct::\_dateOffset, AIRTSP::LegStruct::\_elapsed, AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, AIRTSP::FlightPeriodStruct::\_itLeg, AIRTSP::FlightPeriodStruct::\_itSeconds, AIRTSP::LegStruct::\_offDateOffset, and AIRTSP::FlightPeriodStruct::getTime().

## 10.80.4 Member Data Documentation

### 10.80.4.1 [FlightPeriodStruct&](#) AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [[inherited](#)]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::doEndFlight::operator(), AIRTSP::ScheduleParserHelper::storeFClasses::operator(), AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

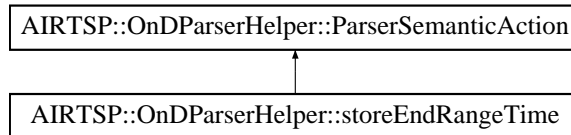
The documentation for this struct was generated from the following files:

- airtsp/command/[ScheduleParserHelper.hpp](#)
- airtsp/command/[ScheduleParserHelper.cpp](#)

## 10.81 AIRTSP::OnDParserHelper::storeEndRangeTime Struct Reference

```
#include <airtsp/command/OnDParserHelper.hpp>
```

Inheritance diagram for AIRTSP::OnDParserHelper::storeEndRangeTime::



### Public Member Functions

- [storeEndRangeTime](#) ([OnDPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [OnDPeriodStruct](#) & [\\_onDPeriod](#)

#### 10.81.1 Detailed Description

Store the end range time.

Definition at line 82 of file OnDParserHelper.hpp.

#### 10.81.2 Constructor & Destructor Documentation

##### 10.81.2.1 AIRTSP::OnDParserHelper::storeEndRangeTime::storeEndRangeTime ([OnDPeriodStruct](#) &)

Actor Constructor.

Definition at line 124 of file OnDParserHelper.cpp.

#### 10.81.3 Member Function Documentation

##### 10.81.3.1 void AIRTSP::OnDParserHelper::storeEndRangeTime::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 129 of file OnDParserHelper.cpp.

References [AIRTSP::OnDPeriodStruct::\\_itSeconds](#), [AIRTSP::OnDParserHelper::ParserSemanticAction::\\_onDPeriod](#), [AIRTSP::OnDPeriodStruct::\\_timeRangeEnd](#), and [AIRTSP::OnDPeriodStruct::getTime\(\)](#).

#### 10.81.4 Member Data Documentation

##### 10.81.4.1 [OnDPeriodStruct](#)& [AIRTSP::OnDParserHelper::ParserSemanticAction::\\_onDPeriod](#) [inherited]

Actor Context.

Definition at line 38 of file OnDParserHelper.hpp.

Referenced by AIRTSP::OnDParserHelper::doEndOnD::operator(), AIRTSP::OnDParserHelper::storeClassCode::operator(), AIRTSP::OnDParserHelper::storeAirlineCode::operator(), operator(), AIRTSP::OnDParserHelper::storeStartRangeTime::operator(), AIRTSP::OnDParserHelper::storeDateRangeEnd::operator(), AIRTSP::OnDParserHelper::storeDateRangeStart::operator(), AIRTSP::OnDParserHelper::storeDestination::operator(), and AIRTSP::OnDParserHelper::storeOrigin::operator().

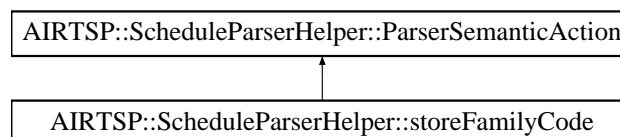
The documentation for this struct was generated from the following files:

- airtsp/command/OnDParserHelper.hpp
- airtsp/command/OnDParserHelper.cpp

## 10.82 AIRTSP::ScheduleParserHelper::storeFamilyCode Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeFamilyCode::



### Public Member Functions

- storeFamilyCode (FlightPeriodStruct &)
- void operator() (int iCode) const

### Public Attributes

- FlightPeriodStruct & \_flightPeriod

### 10.82.1 Detailed Description

Store the parsed family code.

Definition at line 192 of file ScheduleParserHelper.hpp.

### 10.82.2 Constructor & Destructor Documentation

#### 10.82.2.1 AIRTSP::ScheduleParserHelper::storeFamilyCode::storeFamilyCode (FlightPeriodStruct &)

Actor Constructor.

Definition at line 370 of file ScheduleParserHelper.cpp.



### 10.82.3 Member Function Documentation

#### 10.82.3.1 void AIRTSP::ScheduleParserHelper::storeFamilyCode::operator() (int iCode) const

Actor Function (functor).

Definition at line 375 of file ScheduleParserHelper.cpp.

References AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, AIRTSP::SegmentCabinStruct::\_itFamilyCode, and AIRTSP::FlightPeriodStruct::\_itSegmentCabin.

### 10.82.4 Member Data Documentation

#### 10.82.4.1 FlightPeriodStruct& AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::doEndFlight::operator(), AIRTSP::ScheduleParserHelper::storeFClasses::operator(), AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator(), operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

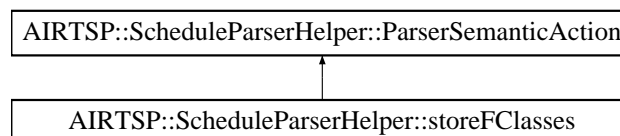
The documentation for this struct was generated from the following files:

- airtsp/command/ScheduleParserHelper.hpp
- airtsp/command/ScheduleParserHelper.cpp

## 10.83 AIRTSP::ScheduleParserHelper::storeFClasses Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeFClasses::



**Public Member Functions**

- [storeFClasses](#) ([FlightPeriodStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

**Public Attributes**

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

**10.83.1 Detailed Description**

Store the parsed list of class codes (for families).

Definition at line 216 of file `ScheduleParserHelper.hpp`.

**10.83.2 Constructor & Destructor Documentation****10.83.2.1 AIRTSP::ScheduleParserHelper::storeFClasses::storeFClasses ([FlightPeriodStruct](#) &)**

Actor Constructor.

Definition at line 410 of file `ScheduleParserHelper.cpp`.

**10.83.3 Member Function Documentation****10.83.3.1 void AIRTSP::ScheduleParserHelper::storeFClasses::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const**

Actor Function (functor).

Definition at line 415 of file `ScheduleParserHelper.cpp`.

References `AIRTSP::FlightPeriodStruct::_areSegmentDefinitionsSpecific`, `AIRTSP::ScheduleParserHelper::ParserSemanticAction::_flightPeriod`, `AIRTSP::SegmentCabinStruct::_itFamilyCode`, `AIRTSP::SegmentCabinStruct::_itFFDisutilityCurveKey`, `AIRTSP::SegmentCabinStruct::_itFRAT5CurveKey`, `AIRTSP::FlightPeriodStruct::_itSegment`, `AIRTSP::FlightPeriodStruct::_itSegmentCabin`, and `AIRTSP::FlightPeriodStruct::addFareFamily()`.

**10.83.4 Member Data Documentation****10.83.4.1 [FlightPeriodStruct&](#) AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [inherited]**

Actor Context.

Definition at line 33 of file `ScheduleParserHelper.hpp`.

Referenced by `AIRTSP::ScheduleParserHelper::doEndFlight::operator()`, `operator()`, `AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator()`, `AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator()`, `AIRTSP::ScheduleParserHelper::storeFamilyCode::operator()`, `AIRTSP::ScheduleParserHelper::storeClasses::operator()`, `AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator()`, `AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator()`, and `AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator()`.

AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

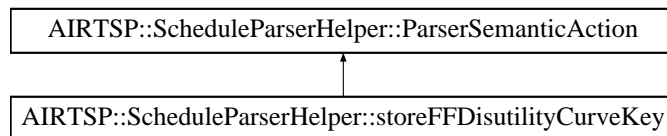
The documentation for this struct was generated from the following files:

- airtsp/command/[ScheduleParserHelper.hpp](#)
- airtsp/command/[ScheduleParserHelper.cpp](#)

## 10.84 AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::



### Public Member Functions

- [storeFFDisutilityCurveKey](#) ([FlightPeriodStruct](#) &)
- [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.84.1 Detailed Description

Store the FFDisutility curve key.

Definition at line 208 of file [ScheduleParserHelper.hpp](#).

#### 10.84.2 Constructor & Destructor Documentation

##### 10.84.2.1 AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::storeFFDisutilityCurveKey ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 397 of file ScheduleParserHelper.cpp.

### 10.84.3 Member Function Documentation

**10.84.3.1** void AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator()  
(iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 402 of file ScheduleParserHelper.cpp.

References AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, AIRTSP::SegmentCabinStruct::\_itFFDisutilityCurveKey, and AIRTSP::FlightPeriodStruct::\_itSegmentCabin.

### 10.84.4 Member Data Documentation

**10.84.4.1** FlightPeriodStruct& AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flight-  
Period [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::doEndFlight::operator(), AIRTSP::ScheduleParserHelper::storeFCClasses::operator(), operator(), AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

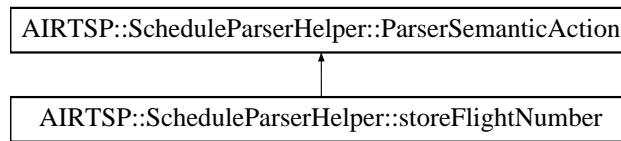
The documentation for this struct was generated from the following files:

- airtsp/command/ScheduleParserHelper.hpp
- airtsp/command/ScheduleParserHelper.cpp

## 10.85 AIRTSP::ScheduleParserHelper::storeFlightNumber Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeFlightNumber:



### Public Member Functions

- [storeFlightNumber](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) (unsigned int iNumber) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

### 10.85.1 Detailed Description

Store the parsed flight number.

Definition at line 45 of file `ScheduleParserHelper.hpp`.

### 10.85.2 Constructor & Destructor Documentation

#### 10.85.2.1 AIRTSP::ScheduleParserHelper::storeFlightNumber::storeFlightNumber ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 50 of file `ScheduleParserHelper.cpp`.

### 10.85.3 Member Function Documentation

#### 10.85.3.1 void AIRTSP::ScheduleParserHelper::storeFlightNumber::operator() (unsigned int *iNumber*) const

Actor Function (functor).

Definition at line 55 of file `ScheduleParserHelper.cpp`.

References `AIRTSP::FlightPeriodStruct::_flightNumber`, and `AIRTSP::ScheduleParserHelper::ParserSemanticAction::_flightPeriod`.

### 10.85.4 Member Data Documentation

#### 10.85.4.1 [FlightPeriodStruct](#)& [AIRTSP::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#) [inherited]

Actor Context.

Definition at line 33 of file `ScheduleParserHelper.hpp`.

Referenced by `AIRTSP::ScheduleParserHelper::doEndFlight::operator()`, `AIRTSP::ScheduleParserHelper::storeFCclasses::operator()`, `AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator()`, and `AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator()`.

AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

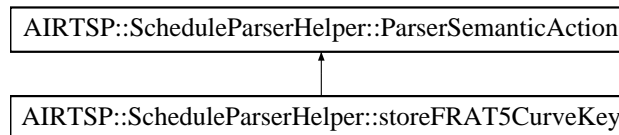
The documentation for this struct was generated from the following files:

- airtsp/command/[ScheduleParserHelper.hpp](#)
- airtsp/command/[ScheduleParserHelper.cpp](#)

## 10.86 AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::



### Public Member Functions

- [storeFRAT5CurveKey](#) ([FlightPeriodStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.86.1 Detailed Description

Store the FRAT5 curve key.

Definition at line 200 of file [ScheduleParserHelper.hpp](#).

#### 10.86.2 Constructor & Destructor Documentation

##### 10.86.2.1 AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::storeFRAT5CurveKey ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 383 of file ScheduleParserHelper.cpp.

### 10.86.3 Member Function Documentation

#### 10.86.3.1 void AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator() (iterator\_t i-Str, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 388 of file ScheduleParserHelper.cpp.

References AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, AIRTSP::SegmentCabinStruct::\_itFRAT5CurveKey, and AIRTSP::FlightPeriodStruct::\_itSegmentCabin.

### 10.86.4 Member Data Documentation

#### 10.86.4.1 FlightPeriodStruct& AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flight-Period [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::doEndFlight::operator(), AIRTSP::ScheduleParserHelper::storeFClasses::operator(), AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator(), operator(), AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

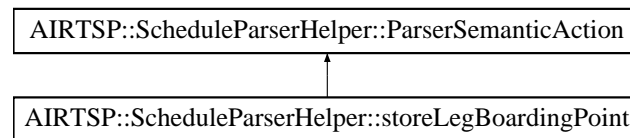
The documentation for this struct was generated from the following files:

- airtsp/command/ScheduleParserHelper.hpp
- airtsp/command/ScheduleParserHelper.cpp

## 10.87 AIRTSP::ScheduleParserHelper::storeLegBoardingPoint Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::



## Public Member Functions

- [storeLegBoardingPoint](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

## Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

### 10.87.1 Detailed Description

Store the parsed leg boarding point.

Definition at line 77 of file `ScheduleParserHelper.hpp`.

### 10.87.2 Constructor & Destructor Documentation

#### 10.87.2.1 AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::storeLegBoardingPoint ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 117 of file `ScheduleParserHelper.cpp`.

### 10.87.3 Member Function Documentation

#### 10.87.3.1 void AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 122 of file `ScheduleParserHelper.cpp`.

References `AIRTSP::LegStruct::_boardingPoint`, `AIRTSP::LegStruct::_cabinList`, `AIRTSP::ScheduleParserHelper::ParserSemanticAction::_flightPeriod`, `AIRTSP::FlightPeriodStruct::_itLeg`, `AIRTSP::FlightPeriodStruct::_legAlreadyDefined`, `AIRTSP::FlightPeriodStruct::_legList`, and `AIRTSP::FlightPeriodStruct::addAirport()`.

### 10.87.4 Member Data Documentation

#### 10.87.4.1 [FlightPeriodStruct](#)& [AIRTSP::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#) [inherited]

Actor Context.

Definition at line 33 of file `ScheduleParserHelper.hpp`.



Referenced by AIRTSP::ScheduleParserHelper::doEndFlight::operator(), AIRTSP::ScheduleParserHelper::storeFClasses::operator(), AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

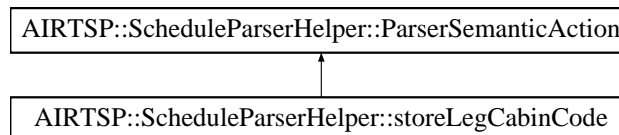
The documentation for this struct was generated from the following files:

- airtsp/command/[ScheduleParserHelper.hpp](#)
- airtsp/command/[ScheduleParserHelper.cpp](#)

## 10.88 AIRTSP::ScheduleParserHelper::storeLegCabinCode Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeLegCabinCode::



### Public Member Functions

- [storeLegCabinCode](#) ([FlightPeriodStruct](#) &)
- [operator\(\)](#) (char iChar) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

### 10.88.1 Detailed Description

Store the parsed leg cabin code.

Definition at line 133 of file [ScheduleParserHelper.hpp](#).

## 10.88.2 Constructor & Destructor Documentation

### 10.88.2.1 AIRTSP::ScheduleParserHelper::storeLegCabinCode::storeLegCabinCode (FlightPeriodStruct &)

Actor Constructor.

Definition at line 251 of file ScheduleParserHelper.cpp.

## 10.88.3 Member Function Documentation

### 10.88.3.1 void AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator() (char iChar) const

Actor Function (functor).

Definition at line 256 of file ScheduleParserHelper.cpp.

References AIRTSP::LegCabinStruct::\_cabinCode, AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, and AIRTSP::FlightPeriodStruct::\_itLegCabin.

## 10.88.4 Member Data Documentation

### 10.88.4.1 FlightPeriodStruct& AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::doEndFlight::operator(), AIRTSP::ScheduleParserHelper::storeFCClasses::operator(), AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

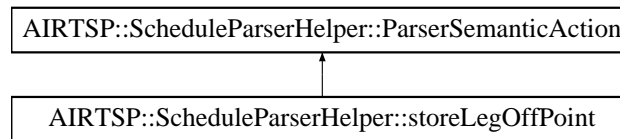
The documentation for this struct was generated from the following files:

- airtsp/command/ScheduleParserHelper.hpp
- airtsp/command/ScheduleParserHelper.cpp

## 10.89 AIRTSP::ScheduleParserHelper::storeLegOffPoint Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeLegOffPoint::



## Public Member Functions

- [storeLegOffPoint](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

## Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

### 10.89.1 Detailed Description

Store the parsed leg off point.

Definition at line 85 of file ScheduleParserHelper.hpp.

### 10.89.2 Constructor & Destructor Documentation

#### 10.89.2.1 AIRTSP::ScheduleParserHelper::storeLegOffPoint::storeLegOffPoint ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 146 of file ScheduleParserHelper.cpp.

### 10.89.3 Member Function Documentation

#### 10.89.3.1 void AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 151 of file ScheduleParserHelper.cpp.

References [AIRTSP::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#), [AIRTSP::FlightPeriodStruct::\\_itLeg](#), [AIRTSP::LegStruct::\\_offPoint](#), and [AIRTSP::FlightPeriodStruct::addAirport\(\)](#).

### 10.89.4 Member Data Documentation

#### 10.89.4.1 [FlightPeriodStruct](#)& [AIRTSP::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#) [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::doEndFlight::operator(), AIRTSP::ScheduleParserHelper::storeFClasses::operator(), AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

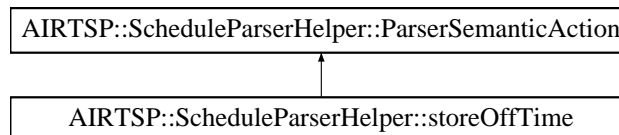
The documentation for this struct was generated from the following files:

- airtsp/command/[ScheduleParserHelper.hpp](#)
- airtsp/command/[ScheduleParserHelper.cpp](#)

## 10.90 AIRTSP::ScheduleParserHelper::storeOffTime Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeOffTime::



### Public Member Functions

- [storeOffTime](#) ([FlightPeriodStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.90.1 Detailed Description

Store the off time.

Definition at line 117 of file [ScheduleParserHelper.hpp](#).

## 10.90.2 Constructor & Destructor Documentation

### 10.90.2.1 AIRTSP::ScheduleParserHelper::storeOffTime::storeOffTime ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 209 of file ScheduleParserHelper.cpp.

## 10.90.3 Member Function Documentation

### 10.90.3.1 void AIRTSP::ScheduleParserHelper::storeOffTime::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 214 of file ScheduleParserHelper.cpp.

References AIRTSP::LegStruct::\_boardingDateOffset, AIRTSP::FlightPeriodStruct::\_dateOffset, AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, AIRTSP::FlightPeriodStruct::\_itLeg, AIRTSP::FlightPeriodStruct::\_itSeconds, AIRTSP::LegStruct::\_offTime, and AIRTSP::FlightPeriodStruct::getTime().

## 10.90.4 Member Data Documentation

### 10.90.4.1 [FlightPeriodStruct&](#) AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flight-Period [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::doEndFlight::operator(), AIRTSP::ScheduleParserHelper::storeFCClasses::operator(), AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

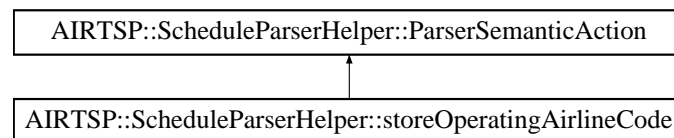
The documentation for this struct was generated from the following files:

- airtsp/command/[ScheduleParserHelper.hpp](#)
- airtsp/command/[ScheduleParserHelper.cpp](#)

## 10.91 AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::



### Public Member Functions

- [storeOperatingAirlineCode](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.91.1 Detailed Description

Store the parsed operating airline code.

Definition at line 93 of file ScheduleParserHelper.hpp.

#### 10.91.2 Constructor & Destructor Documentation

##### 10.91.2.1 AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::storeOperatingAirlineCode ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 162 of file ScheduleParserHelper.cpp.

#### 10.91.3 Member Function Documentation

##### 10.91.3.1 void AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 167 of file ScheduleParserHelper.cpp.

References [AIRTSP::LegStruct::\\_airlineCode](#), [AIRTSP::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#), and [AIRTSP::FlightPeriodStruct::\\_itLeg](#).

#### 10.91.4 Member Data Documentation

##### 10.91.4.1 [FlightPeriodStruct](#)& AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::doEndFlight::operator(), AIRTSP::ScheduleParserHelper::storeFCClasses::operator(), AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

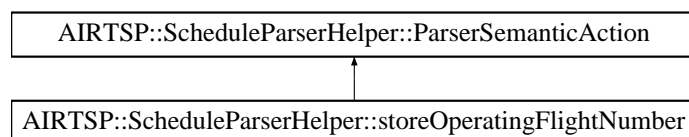
The documentation for this struct was generated from the following files:

- airtsp/command/ScheduleParserHelper.hpp
- airtsp/command/ScheduleParserHelper.cpp

## 10.92 AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::



### Public Member Functions

- [storeOperatingFlightNumber](#) ([FlightPeriodStruct](#) &)
- [operator\(\)](#) (unsigned int iNumber) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.92.1 Detailed Description

Store the parsed operating flight number.

Definition at line 101 of file ScheduleParserHelper.hpp.

## 10.92.2 Constructor & Destructor Documentation

### 10.92.2.1 AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::storeOperatingFlightNumber (FlightPeriodStruct &)

Actor Constructor.

Definition at line 179 of file ScheduleParserHelper.cpp.

## 10.92.3 Member Function Documentation

### 10.92.3.1 void AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator() (unsigned int iNumber) const

Actor Function (functor).

Definition at line 184 of file ScheduleParserHelper.cpp.

References AIRTSP::LegStruct::\_flightNumber, AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, and AIRTSP::FlightPeriodStruct::\_itLeg.

## 10.92.4 Member Data Documentation

### 10.92.4.1 FlightPeriodStruct& AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::doEndFlight::operator(), AIRTSP::ScheduleParserHelper::storeFCClasses::operator(), AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

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

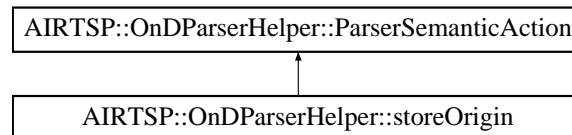
- airtsp/command/ScheduleParserHelper.hpp
- airtsp/command/ScheduleParserHelper.cpp



## 10.93 AIRTSP::OnDParserHelper::storeOrigin Struct Reference

```
#include <airtsp/command/OnDParserHelper.hpp>
```

Inheritance diagram for AIRTSP::OnDParserHelper::storeOrigin::



### Public Member Functions

- [storeOrigin](#) ([OnDPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [OnDPeriodStruct](#) & [\\_onDPeriod](#)

#### 10.93.1 Detailed Description

Store the parsed origin.

Definition at line 42 of file OnDParserHelper.hpp.

#### 10.93.2 Constructor & Destructor Documentation

##### 10.93.2.1 AIRTSP::OnDParserHelper::storeOrigin::storeOrigin ([OnDPeriodStruct](#) &)

Actor Constructor.

Definition at line 30 of file OnDParserHelper.cpp.

#### 10.93.3 Member Function Documentation

##### 10.93.3.1 void AIRTSP::OnDParserHelper::storeOrigin::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStr-End) const

Actor Function (functor).

Definition at line 35 of file OnDParserHelper.cpp.

References [AIRTSP::OnDPeriodStruct::\\_airlineCode](#), [AIRTSP::OnDPeriodStruct::\\_airlineCodeList](#), [AIRTSP::OnDPeriodStruct::\\_classCode](#), [AIRTSP::OnDPeriodStruct::\\_classCodeList](#), [AIRTSP::OnDPeriodStruct::\\_nbOfAirlines](#), [AIRTSP::OnDParserHelper::ParserSemanticAction::\\_onDPeriod](#), and [AIRTSP::OnDPeriodStruct::\\_origin](#).

#### 10.93.4 Member Data Documentation

##### 10.93.4.1 [OnDPeriodStruct](#)& [AIRTSP::OnDParserHelper::ParserSemanticAction::\\_onDPeriod](#) [inherited]

Actor Context.

Definition at line 38 of file OnDParserHelper.hpp.

Referenced by AIRTSP::OnDParserHelper::doEndOnD::operator(), AIRTSP::OnDParserHelper::storeClassCode::operator(), AIRTSP::OnDParserHelper::storeAirlineCode::operator(), AIRTSP::OnDParserHelper::storeEndRangeTime::operator(), AIRTSP::OnDParserHelper::storeStartRangeTime::operator(), AIRTSP::OnDParserHelper::storeDateRangeEnd::operator(), AIRTSP::OnDParserHelper::storeDateRangeStart::operator(), AIRTSP::OnDParserHelper::storeDestination::operator(), and operator().

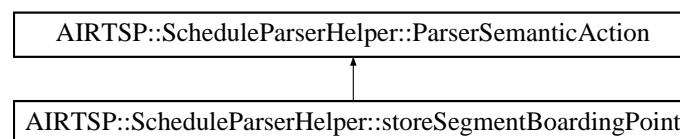
The documentation for this struct was generated from the following files:

- airtsp/command/OnDParserHelper.hpp
- airtsp/command/OnDParserHelper.cpp

## 10.94 AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::



### Public Member Functions

- [storeSegmentBoardingPoint](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.94.1 Detailed Description

Store the parsed segment boarding point.

Definition at line 160 of file ScheduleParserHelper.hpp.

#### 10.94.2 Constructor & Destructor Documentation

##### 10.94.2.1 AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::storeSegmentBoardingPoint ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 308 of file ScheduleParserHelper.cpp.

### 10.94.3 Member Function Documentation

#### 10.94.3.1 void AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 313 of file ScheduleParserHelper.cpp.

References AIRTSP::SegmentStruct::\_boardingPoint, AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, and AIRTSP::FlightPeriodStruct::\_itSegment.

### 10.94.4 Member Data Documentation

#### 10.94.4.1 FlightPeriodStruct& AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flight- Period [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::doEndFlight::operator(), AIRTSP::ScheduleParserHelper::storeFClasses::operator(), AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator(), operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

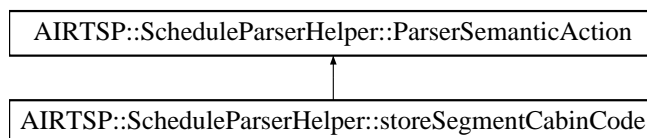
The documentation for this struct was generated from the following files:

- airtsp/command/ScheduleParserHelper.hpp
- airtsp/command/ScheduleParserHelper.cpp

## 10.95 AIRTSP::ScheduleParserHelper::storeSegmentCabinCode Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::



## Public Member Functions

- [storeSegmentCabinCode](#) ([FlightPeriodStruct](#) &)
- void [operator\(\)](#) (char iChar) const

## Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

### 10.95.1 Detailed Description

Store the parsed segment cabin code.

Definition at line 176 of file ScheduleParserHelper.hpp.

### 10.95.2 Constructor & Destructor Documentation

#### 10.95.2.1 AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::storeSegmentCabinCode ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 334 of file ScheduleParserHelper.cpp.

### 10.95.3 Member Function Documentation

#### 10.95.3.1 void AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator() (char *iChar*) const

Actor Function (functor).

Definition at line 339 of file ScheduleParserHelper.cpp.

References [AIRTSP::SegmentCabinStruct::\\_cabinCode](#), [AIRTSP::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#), and [AIRTSP::FlightPeriodStruct::\\_itSegmentCabin](#).

### 10.95.4 Member Data Documentation

#### 10.95.4.1 [FlightPeriodStruct](#)& [AIRTSP::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#) [*inherited*]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

Referenced by [AIRTSP::ScheduleParserHelper::doEndFlight::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeFClasses::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeFamilyCode::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeClasses::operator\(\)](#), [operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeCapacity::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeElapsedTime::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeOffTime::operator\(\)](#), and [AIRTSP::ScheduleParserHelper::storeBoardingTime::operator\(\)](#).

AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator>(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator>(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator>(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator>(), AIRTSP::ScheduleParserHelper::storeDow::operator>(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator>(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator>(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator()).

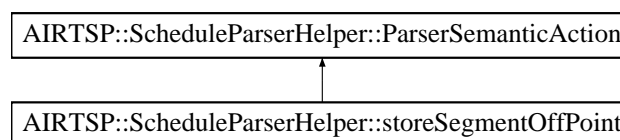
The documentation for this struct was generated from the following files:

- airtsp/command/[ScheduleParserHelper.hpp](#)
- airtsp/command/[ScheduleParserHelper.cpp](#)

## 10.96 AIRTSP::ScheduleParserHelper::storeSegmentOffPoint Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::



### Public Member Functions

- [storeSegmentOffPoint](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.96.1 Detailed Description

Store the parsed segment off point.

Definition at line 168 of file [ScheduleParserHelper.hpp](#).

#### 10.96.2 Constructor & Destructor Documentation

##### 10.96.2.1 AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::storeSegmentOffPoint ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 321 of file [ScheduleParserHelper.cpp](#).

### 10.96.3 Member Function Documentation

#### 10.96.3.1 void AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 326 of file ScheduleParserHelper.cpp.

References AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, AIRTSP::FlightPeriodStruct::\_itSegment, and AIRTSP::SegmentStruct::\_offPoint.

### 10.96.4 Member Data Documentation

#### 10.96.4.1 FlightPeriodStruct& AIRTSP::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

Referenced by AIRTSP::ScheduleParserHelper::doEndFlight::operator(), AIRTSP::ScheduleParserHelper::storeFClasses::operator(), AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator(), AIRTSP::ScheduleParserHelper::storeFamilyCode::operator(), AIRTSP::ScheduleParserHelper::storeClasses::operator(), AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator(), operator(), AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRTSP::ScheduleParserHelper::storeCapacity::operator(), AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator(), AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

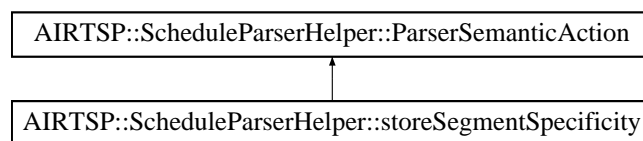
The documentation for this struct was generated from the following files:

- airtsp/command/ScheduleParserHelper.hpp
- airtsp/command/ScheduleParserHelper.cpp

## 10.97 AIRTSP::ScheduleParserHelper::storeSegmentSpecificity Struct Reference

```
#include <airtsp/command/ScheduleParserHelper.hpp>
```

Inheritance diagram for AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::



**Public Member Functions**

- [storeSegmentSpecificity](#) ([FlightPeriodStruct](#) &)
- void [operator\(\)](#) (char iChar) const

**Public Attributes**

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

**10.97.1 Detailed Description**

Store whether or not the segment definitions are specific. Specific means that there is a definition for each segment. General (not specific) means that a single definition defines all the segments.

Definition at line 152 of file ScheduleParserHelper.hpp.

**10.97.2 Constructor & Destructor Documentation****10.97.2.1 AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::storeSegmentSpecificity ([FlightPeriodStruct](#) &)**

Actor Constructor.

Definition at line 282 of file ScheduleParserHelper.cpp.

**10.97.3 Member Function Documentation****10.97.3.1 void AIRTSP::ScheduleParserHelper::storeSegmentSpecificity::operator() (char iChar) const**

Actor Function (functor).

Definition at line 287 of file ScheduleParserHelper.cpp.

References [AIRTSP::FlightPeriodStruct::\\_airportList](#), [AIRTSP::FlightPeriodStruct::\\_airportOrderedList](#), [AIRTSP::FlightPeriodStruct::\\_areSegmentDefinitionsSpecific](#), [AIRTSP::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#), and [AIRTSP::FlightPeriodStruct::buildSegments\(\)](#).

**10.97.4 Member Data Documentation****10.97.4.1 [FlightPeriodStruct&](#) [AIRTSP::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#) [inherited]**

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

Referenced by [AIRTSP::ScheduleParserHelper::doEndFlight::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeFClasses::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeFamilyCode::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeClasses::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeSegmentCabinCode::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeSegmentOffPoint::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint::operator\(\)](#), [operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeCapacity::operator\(\)](#), [AIRTSP::ScheduleParserHelper::storeLegCabinCode::operator\(\)](#),

AIRTSP::ScheduleParserHelper::storeElapsedTime::operator(), AIRTSP::ScheduleParserHelper::storeOffTime::operator(), AIRTSP::ScheduleParserHelper::storeBoardingTime::operator(), AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRTSP::ScheduleParserHelper::storeLegOffPoint::operator(), AIRTSP::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRTSP::ScheduleParserHelper::storeDow::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRTSP::ScheduleParserHelper::storeDateRangeStart::operator(), AIRTSP::ScheduleParserHelper::storeFlightNumber::operator(), and AIRTSP::ScheduleParserHelper::storeAirlineCode::operator().

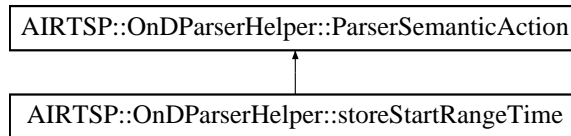
The documentation for this struct was generated from the following files:

- airtsp/command/[ScheduleParserHelper.hpp](#)
- airtsp/command/[ScheduleParserHelper.cpp](#)

## 10.98 AIRTSP::OnDParserHelper::storeStartRangeTime Struct Reference

```
#include <airtsp/command/OnDParserHelper.hpp>
```

Inheritance diagram for AIRTSP::OnDParserHelper::storeStartRangeTime::



### Public Member Functions

- [storeStartRangeTime](#) (OnDPeriodStruct &)
- void [operator\(\)](#) (iterator\_t iStr, iterator\_t iStrEnd) const

### Public Attributes

- [OnDPeriodStruct](#) & [\\_onDPeriod](#)

#### 10.98.1 Detailed Description

Store the start range time.

Definition at line 74 of file OnDParserHelper.hpp.

#### 10.98.2 Constructor & Destructor Documentation

##### 10.98.2.1 AIRTSP::OnDParserHelper::storeStartRangeTime::storeStartRangeTime ([OnDPeriodStruct](#) &)

Actor Constructor.

Definition at line 109 of file OnDParserHelper.cpp.



### 10.98.3 Member Function Documentation

#### 10.98.3.1 void AIRTSP::OnDParserHelper::storeStartRangeTime::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 114 of file OnDParserHelper.cpp.

References AIRTSP::OnDPeriodStruct::\_itSeconds, AIRTSP::OnDParserHelper::ParserSemanticAction::\_onDPeriod, AIRTSP::OnDPeriodStruct::\_timeRangeStart, and AIRTSP::OnDPeriodStruct::getTime().

### 10.98.4 Member Data Documentation

#### 10.98.4.1 OnDPeriodStruct& AIRTSP::OnDParserHelper::ParserSemanticAction::\_onDPeriod [inherited]

Actor Context.

Definition at line 38 of file OnDParserHelper.hpp.

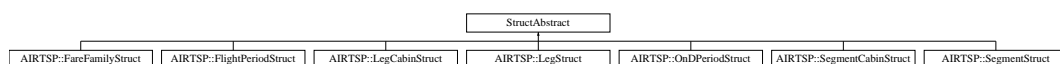
Referenced by AIRTSP::OnDParserHelper::doEndOnD::operator(), AIRTSP::OnDParserHelper::storeClassCode::operator(), AIRTSP::OnDParserHelper::storeAirlineCode::operator(), AIRTSP::OnDParserHelper::storeEndRangeTime::operator(), operator(), AIRTSP::OnDParserHelper::storeDateRangeEnd::operator(), AIRTSP::OnDParserHelper::storeDateRangeStart::operator(), AIRTSP::OnDParserHelper::storeDestination::operator(), and AIRTSP::OnDParserHelper::storeOrigin::operator().

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

- [airtsp/command/OnDParserHelper.hpp](#)
- [airtsp/command/OnDParserHelper.cpp](#)

## 10.99 StructAbstract Class Reference

Inheritance diagram for StructAbstract::



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

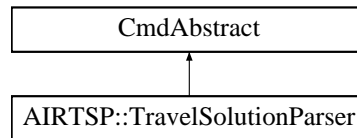
- [airtsp/bom/SegmentCabinStruct.hpp](#)
- [airtsp/bom/LegStruct.hpp](#)
- [airtsp/bom/FareFamilyStruct.hpp](#)
- [airtsp/bom/FlightPeriodStruct.hpp](#)
- [airtsp/bom/LegCabinStruct.hpp](#)
- [airtsp/bom/OnDPeriodStruct.hpp](#)
- [airtsp/bom/SegmentStruct.hpp](#)

## 10.100 AIRTSP::TravelSolutionParser Class Reference

Class filling the TravelSolutionHolder structure (representing a list of classes/travelSolutions) from a given input file.

```
#include <airtsp/command/TravelSolutionParser.hpp>
```

Inheritance diagram for AIRTSP::TravelSolutionParser::



### Static Public Member Functions

- static bool [parseInputFileAndBuildBom](#) (const stdair::Filename\_T &)

#### 10.100.1 Detailed Description

Class filling the TravelSolutionHolder structure (representing a list of classes/travelSolutions) from a given input file.

Definition at line 19 of file TravelSolutionParser.hpp.

#### 10.100.2 Member Function Documentation

**10.100.2.1 static bool AIRTSP::TravelSolutionParser::parseInputFileAndBuildBom (const stdair::Filename\_T &) [static]**

Parse the input values from a CSV-formatted travel solution file.

##### Parameters:

*const* std::string& inputFileNames Travel solution file to be parsed.

##### Returns:

bool Whether or not the parsing was successful.

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

- airtsp/command/[TravelSolutionParser.hpp](#)

## 11 AirTSP File Documentation

### 11.1 airtsp/AIRTSP\_Service.hpp File Reference

```
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_service_types.hpp>
#include <stdair/stdair_file.hpp>
#include <stdair/bom/TravelSolutionTypes.hpp>
```

**Namespaces**

- namespace [stdair](#)
- namespace [AIRTSP](#)

**Classes**

- class [AIRTSP::AIRTSP\\_Service](#)  
*Interface for the Airtsp Services.*

**11.2 airtsp/AIRTSP\_Types.hpp File Reference**

```
#include <boost/shared_ptr.hpp>
#include <stdair/stdair_exceptions.hpp>
```

**Namespaces**

- namespace [AIRTSP](#)

**Classes**

- class [AIRTSP::SegmentDateNotFoundException](#)
- class [AIRTSP::OnDInputFileNotFoundException](#)
- class [AIRTSP::ScheduleInputFileNotFoundException](#)

**Typedefs**

- typedef boost::shared\_ptr< AIRTSP\_Service > [AIRTSP::AIRTSP\\_ServicePtr\\_T](#)

**11.3 airtsp/basic/BasConst.cpp File Reference**

```
#include <airtsp/basic/BasConst_General.hpp>
#include <airtsp/basic/BasConst_AIRTSP_Service.hpp>
```

**Namespaces**

- namespace [AIRTSP](#)

**Functions**

- const stdair::Duration\_T [AIRTSP::MINIMUM\\_TIME\\_BETWEEN\\_REQUEST\\_AND\\_DEPARTURE](#) (4, 0, 0)

**Variables**

- const int [AIRTSP::DEFAULT\\_NUMBER\\_OF\\_DRAWS\\_FOR\\_MC\\_SIMULATION](#) = 100000

## 11.4 airtsp/basic/BasConst\_AIRTSP\_Service.hpp File Reference

### Namespaces

- namespace [AIRTSP](#)

## 11.5 airtsp/basic/BasConst\_General.hpp File Reference

```
#include <stdair/stdair_date_time_types.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

### Variables

- const int [AIRTSP::DEFAULT\\_NUMBER\\_OF\\_DRAWS\\_FOR\\_MC\\_SIMULATION](#)
- const stdair::Duration\_T [AIRTSP::MINIMUM\\_TIME\\_BETWEEN\\_REQUEST\\_AND\\_DEPARTURE](#)

## 11.6 airtsp/basic/BasParserTypes.hpp File Reference

```
#include <string>
#include <boost/spirit/home/classic/core.hpp>
#include <boost/spirit/home/classic/attribute.hpp>
#include <boost/spirit/home/classic/utility/functor_parser.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>
#include <boost/spirit/home/classic/actor/push_back_actor.hpp>
#include <boost/spirit/home/classic/actor/assign_actor.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

### Typedefs

- typedef char [AIRTSP::char\\_t](#)
- typedef boost::spirit::classic::file\_iterator< [char\\_t](#) > [AIRTSP::iterator\\_t](#)
- typedef boost::spirit::classic::scanner< [iterator\\_t](#) > [AIRTSP::scanner\\_t](#)
- typedef boost::spirit::classic::rule< [scanner\\_t](#) > [AIRTSP::rule\\_t](#)
- typedef boost::spirit::classic::int\_parser< unsigned int, 10, 1, 1 > [AIRTSP::int1\\_p\\_t](#)
- typedef boost::spirit::classic::uint\_parser< unsigned int, 10, 2, 2 > [AIRTSP::uint2\\_p\\_t](#)

- typedef boost::spirit::classic::uint\_parser< unsigned int, 10, 4, 4 > [AIRTSP::uint4\\_p\\_t](#)
- typedef boost::spirit::classic::uint\_parser< unsigned int, 10, 1, 4 > [AIRTSP::uint1\\_4\\_p\\_t](#)
- typedef boost::spirit::classic::chset< [char\\_t](#) > [AIRTSP::chset\\_t](#)
- typedef boost::spirit::classic::impl::loop\_traits< [chset\\_t](#), unsigned int, unsigned int >::type [AIRTSP::repeat\\_p\\_t](#)
- typedef boost::spirit::classic::bounded< [uint2\\_p\\_t](#), unsigned int > [AIRTSP::bounded2\\_p\\_t](#)
- typedef boost::spirit::classic::bounded< [uint4\\_p\\_t](#), unsigned int > [AIRTSP::bounded4\\_p\\_t](#)
- typedef boost::spirit::classic::bounded< [uint1\\_4\\_p\\_t](#), unsigned int > [AIRTSP::bounded1\\_4\\_p\\_t](#)

## 11.7 airtsp/batches/airtsp.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <fstream>
#include <string>
#include <boost/date_time/posix_time/posix_time.hpp>
#include <boost/date_time/gregorian/gregorian.hpp>
#include <boost/program_options.hpp>
#include <boost/tokenizer.hpp>
#include <boost/lexical_cast.hpp>
#include <stdair/STDAIR_Service.hpp>
#include <stdair/bom/BomDisplay.hpp>
#include <stdair/bom/BookingRequestStruct.hpp>
#include <stdair/bom/TravelSolutionStruct.hpp>
#include <stdair/service/Logger.hpp>
#include <airtsp/AIRTSP_Service.hpp>
#include <airtsp/batches/BookingRequestParser.hpp>
#include <airtsp/config/airtsp-paths.hpp>
```

### Typedefs

- typedef std::vector< std::string > [WordList\\_T](#)

### Functions

- const std::string [K\\_AIRTSP\\_DEFAULT\\_LOG\\_FILENAME](#) ("airtsp.log")
- const std::string [K\\_AIRTSP\\_DEFAULT\\_INPUT\\_FILENAME](#) (STDAIR\_SAMPLE\_DIR"/schedule03.csv")
- const std::string [K\\_AIRTSP\\_DEFAULT\\_BOOKING\\_REQUEST](#) ("NCE BKK NCE 2007-04-21 2007-03-21 08:32:00 C 1 DF RO 5 NONE 10:00:00 2000.0 20.0")
- 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, bool &ioReadBookingRequestFromCmdLine, stdair::Filename\_T &ioInputFilename, std::string &ioLogFilename, std::string &ioBookingRequestString)
- stdair::BookingRequestStruct [parseBookingRequest](#) (const std::string &iRequestOption)
- int [main](#) (int argc, char \*argv[])

## Variables

- const bool [K\\_AIRTSP\\_DEFAULT\\_BUILT\\_IN\\_INPUT](#) = false
- const bool [K\\_AIRTSP\\_DEFAULT\\_BOOKING\\_REQUEST\\_MODE](#) = false
- const int [K\\_AIRTSP\\_EARLY\\_RETURN\\_STATUS](#) = 99

### 11.7.1 Typedef Documentation

#### 11.7.1.1 typedef std::vector<std::string> [WordList\\_T](#)

Definition at line 24 of file airtsp.cpp.

### 11.7.2 Function Documentation

#### 11.7.2.1 const std::string [K\\_AIRTSP\\_DEFAULT\\_LOG\\_FILENAME](#) ("airtsp.log")

Default name and location for the log file.

Referenced by [readConfiguration\(\)](#).

#### 11.7.2.2 const std::string [K\\_AIRTSP\\_DEFAULT\\_INPUT\\_FILENAME](#) (STDAIR\_SAMPLE\_DIR"/schedule03.csv")

Default name and location for the (CSV) input file.

Referenced by [readConfiguration\(\)](#).

#### 11.7.2.3 const std::string [K\\_AIRTSP\\_DEFAULT\\_BOOKING\\_REQUEST](#) ("NCE BKK NCE 2007-04-21 2007-03-21 08:32:00 C 1 DF RO 5 NONE 10:00:00 2000.0 20.0")

Default booking request string, to be seached against the AirTSP network.

Referenced by [main\(\)](#).

#### 11.7.2.4 std::string [createStringFromWordList](#) (const [WordList\\_T](#) & *iWordList*)

Definition at line 59 of file airtsp.cpp.

Referenced by [readConfiguration\(\)](#).

#### 11.7.2.5 template<class T> std::ostream& [operator<<](#) (std::ostream & *os*, const std::vector< T > & *v*)

Definition at line 77 of file airtsp.cpp.

#### 11.7.2.6 `int readConfiguration (int argc, char * argv[], bool & ioIsBuiltin, bool & ioReadBookingRequestFromCmdLine, stdair::Filename_T & ioInputFilename, std::string & ioLogFilename, std::string & ioBookingRequestString)`

Read and parse the command line options.

Definition at line 87 of file airtsp.cpp.

References `createStringFromWordList()`, `K_AIRTSP_DEFAULT_BOOKING_REQUEST_MODE`, `K_AIRTSP_DEFAULT_BUILT_IN_INPUT`, `K_AIRTSP_DEFAULT_INPUT_FILENAME()`, `K_AIRTSP_DEFAULT_LOG_FILENAME()`, and `K_AIRTSP_EARLY_RETURN_STATUS`.

Referenced by `main()`.

#### 11.7.2.7 `stdair::BookingRequestStruct parseBookingRequest (const std::string & iRequestOption)`

Definition at line 230 of file airtsp.cpp.

Referenced by `main()`.

#### 11.7.2.8 `int main (int argc, char * argv[ ])`

Definition at line 347 of file airtsp.cpp.

References `AIRTSP::AIRTSP_Service::buildSampleBom()`, `AIRTSP::AIRTSP_Service::buildSegmentPathList()`, `K_AIRTSP_DEFAULT_BOOKING_REQUEST()`, `K_AIRTSP_EARLY_RETURN_STATUS`, `AIRTSP::AIRTSP_Service::parseAndLoad()`, `parseBookingRequest()`, and `readConfiguration()`.

### 11.7.3 Variable Documentation

#### 11.7.3.1 `const bool K_AIRTSP_DEFAULT_BUILT_IN_INPUT = false`

Default for the BOM tree building. The BOM tree can either be built-in or provided by an input file. That latter must then be given with the `-s` option.

Definition at line 44 of file airtsp.cpp.

Referenced by `readConfiguration()`.

#### 11.7.3.2 `const bool K_AIRTSP_DEFAULT_BOOKING_REQUEST_MODE = 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 50 of file airtsp.cpp.

Referenced by `readConfiguration()`.

#### 11.7.3.3 `const int K_AIRTSP_EARLY_RETURN_STATUS = 99`

Early return status (so that it can be differentiated from an error).

Definition at line 84 of file airtsp.cpp.

Referenced by `main()`, and `readConfiguration()`.

## 11.8 airtsp/batches/BookingRequestParser.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <fstream>
#include <boost/date_time/posix_time/posix_time.hpp>
#include <boost/date_time/gregorian/gregorian.hpp>
#include <boost/spirit/home/classic/core.hpp>
#include <boost/spirit/home/classic/attribute.hpp>
#include <boost/spirit/home/classic/utility/functor_parser.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>
#include <boost/spirit/home/classic/actor/push_back_actor.hpp>
#include <boost/spirit/home/classic/actor/assign_actor.hpp>
#include <stdair/service/Logger.hpp>
#include <airtsp/batches/BookingRequestParser.hpp>
```

### Namespaces

- namespace [airtsp](#)
- namespace [boost::spirit::classic](#)

### Classes

- struct [airtsp::store\\_place\\_element](#)
- struct [airtsp::store\\_date](#)
- struct [airtsp::store\\_airline\\_sign](#)
- struct [airtsp::store\\_airline\\_code](#)
- struct [airtsp::store\\_airline\\_name](#)
- struct [airtsp::store\\_passenger\\_number](#)
- struct [airtsp::store\\_adult\\_passenger\\_type](#)
- struct [airtsp::store\\_child\\_passenger\\_type](#)
- struct [airtsp::store\\_pet\\_passenger\\_type](#)
- struct [airtsp::SearchStringParser](#)
- struct [airtsp::SearchStringParser::definition< ScannerT >](#)

### Defines

- #define [BOOST\\_SPIRIT\\_DEBUG](#)



## Typedefs

- typedef char [char\\_t](#)
- typedef char const \* [iterator\\_t](#)
- typedef boost::spirit::classic::scanner< [iterator\\_t](#) > [scanner\\_t](#)
- typedef boost::spirit::classic::rule< [scanner\\_t](#) > [rule\\_t](#)

## Functions

- SearchString\_T [airtsp::parseBookingRequest](#) (const std::string &iSearchString)

## Variables

- boost::spirit::classic::int\_parser< unsigned int, 10, 1, 1 > [airtsp::int1\\_p](#)
- boost::spirit::classic::uint\_parser< unsigned int, 10, 1, 1 > [airtsp::uint1\\_p](#)
- boost::spirit::classic::uint\_parser< unsigned int, 10, 1, 2 > [airtsp::uint1\\_2\\_p](#)
- boost::spirit::classic::uint\_parser< int, 10, 2, 2 > [airtsp::uint2\\_p](#)
- boost::spirit::classic::uint\_parser< int, 10, 2, 4 > [airtsp::uint2\\_4\\_p](#)
- boost::spirit::classic::uint\_parser< int, 10, 4, 4 > [airtsp::uint4\\_p](#)
- boost::spirit::classic::uint\_parser< int, 10, 1, 4 > [airtsp::uint1\\_4\\_p](#)

### 11.8.1 Define Documentation

#### 11.8.1.1 #define BOOST\_SPIRIT\_DEBUG

Definition at line 12 of file BookingRequestParser.cpp.

### 11.8.2 Typedef Documentation

#### 11.8.2.1 typedef char [char\\_t](#)

Definition at line 28 of file BookingRequestParser.cpp.

#### 11.8.2.2 typedef char const\* [iterator\\_t](#)

Definition at line 29 of file BookingRequestParser.cpp.

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

Definition at line 31 of file BookingRequestParser.cpp.

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

Definition at line 32 of file BookingRequestParser.cpp.

## 11.9 airtsp/batches/BookingRequestParser.hpp File Reference

```
#include <string>
#include <vector>
```

## Namespaces

- namespace [airtsp](#)

## Classes

- struct [airtsp::Place\\_T](#)
- struct [airtsp::Date\\_T](#)
- struct [airtsp::Airline\\_T](#)
- struct [airtsp::Passenger\\_T](#)
- struct [airtsp::SearchString\\_T](#)

## Typedefs

- typedef std::vector< Place\_T > [airtsp::PlaceList\\_T](#)
- typedef std::vector< Date\_T > [airtsp::DateList\\_T](#)
- typedef std::vector< Airline\_T > [airtsp::AirlineList\\_T](#)
- typedef std::vector< Passenger\_T > [airtsp::PassengerList\\_T](#)

## Functions

- SearchString\_T [airtsp::parseBookingRequest](#) (const std::string &iSearchString)

## 11.10 airtsp/bom/AirportList.hpp File Reference

```
#include <set>
#include <vector>
#include <stdair/stdair_basic_types.hpp>
```

## Namespaces

- namespace [AIRTSP](#)

## Typedefs

- typedef std::set< stdair::AirportCode\_T > [AIRTSP::AirportList\\_T](#)
- typedef std::vector< stdair::AirportCode\_T > [AIRTSP::AirportOrderedList\\_T](#)

## 11.11 airtsp/bom/BomDisplay.cpp File Reference

```
#include <cassert>
#include <ostream>
#include <stdair/basic/BasConst_BomDisplay.hpp>
#include <stdair/bom/BomManager.hpp>
#include <stdair/bom/BomRoot.hpp>
```

```
#include <airtsp/bom/ReachableUniverse.hpp>
#include <airtsp/bom/BomDisplay.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

### Classes

- struct [AIRTSP::FlagSaver](#)

## 11.12 airtsp/bom/BomDisplay.hpp File Reference

```
#include <iosfwd>
#include <string>
```

### Namespaces

- namespace [stdair](#)
- namespace [AIRTSP](#)

### Classes

- class [AIRTSP::BomDisplay](#)  
*Utility class to display AirTSP objects with a pretty format.*

## 11.13 airtsp/bom/FareFamilyStruct.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <airtsp/bom/FareFamilyStruct.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

## 11.14 airtsp/bom/FareFamilyStruct.hpp File Reference

```
#include <string>
#include <vector>
#include <stdair/stdair_inventory_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

### Classes

- struct [AIRTSP::FareFamilyStruct](#)

### Typedefs

- typedef std::vector< FareFamilyStruct > [AIRTSP::FareFamilyStructList\\_T](#)

## 11.15 airtsp/bom/FlightPeriodStruct.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/basic/BasConst_Period_BOM.hpp>
#include <stdair/service/Logger.hpp>
#include <airtsp/AIRTSP_Types.hpp>
#include <airtsp/bom/FlightPeriodStruct.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

## 11.16 airtsp/bom/FlightPeriodStruct.hpp File Reference

```
#include <string>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
#include <stdair/bom/DoWStruct.hpp>
#include <airtsp/bom/LegCabinStruct.hpp>
#include <airtsp/bom/LegStruct.hpp>
#include <airtsp/bom/SegmentStruct.hpp>
#include <airtsp/bom/SegmentCabinStruct.hpp>
#include <airtsp/bom/FareFamilyStruct.hpp>
#include <airtsp/bom/AirportList.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

## Classes

- struct [AIRTSP::FlightPeriodStruct](#)

## 11.17 airtsp/bom/LegCabinStruct.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/bom/LegCabin.hpp>
#include <airtsp/bom/LegCabinStruct.hpp>
```

## Namespaces

- namespace [AIRTSP](#)

## 11.18 airtsp/bom/LegCabinStruct.hpp File Reference

```
#include <string>
#include <vector>
#include <stdair/stdair_inventory_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
```

## Namespaces

- namespace [stdair](#)
- namespace [AIRTSP](#)

## Classes

- struct [AIRTSP::LegCabinStruct](#)

## Typedefs

- typedef std::vector< LegCabinStruct > [AIRTSP::LegCabinStructList\\_T](#)

## 11.19 airtsp/bom/LegStruct.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/basic/BasConst_Period_BOM.hpp>
#include <stdair/bom/LegDate.hpp>
#include <airtsp/bom/LegStruct.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

## 11.20 airtsp/bom/LegStruct.hpp File Reference

```
#include <string>
#include <vector>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
#include <airtsp/bom/LegCabinStruct.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [AIRTSP](#)

### Classes

- struct [AIRTSP::LegStruct](#)

### Typedefs

- typedef std::vector< LegStruct > [AIRTSP::LegStructList\\_T](#)

## 11.21 airtsp/bom/OnDPeriodStruct.cpp File Reference

```
#include <cassert>
#include <iostream>
#include <stdair/basic/BasConst_Period_BOM.hpp>
#include <stdair/basic/BasConst_General.hpp>
#include <stdair/basic/BasConst_Inventory.hpp>
#include <stdair/service/Logger.hpp>
#include <airtsp/bom/OnDPeriodStruct.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

## 11.22 airtsp/bom/OnDPeriodStruct.hpp File Reference

```
#include <string>
#include <stdair/stdair_inventory_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

### Classes

- struct [AIRTSP::OnDPeriodStruct](#)

## 11.23 airtsp/bom/OriginDestinationSet.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <boost/archive/text_iarchive.hpp>
#include <boost/archive/text_oarchive.hpp>
#include <boost/serialization/access.hpp>
#include <stdair/basic/BasConst_Inventory.hpp>
#include <airtsp/bom/OriginDestinationSet.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

### Functions

- template void [AIRTSP::OriginDestinationSet::serialize< ba::text\\_oarchive >](#) (ba::text\_oarchive &, unsigned int)
- template void [AIRTSP::OriginDestinationSet::serialize< ba::text\\_iarchive >](#) (ba::text\_iarchive &, unsigned int)

## 11.24 airtsp/bom/OriginDestinationSet.hpp File Reference

```
#include <iosfwd>
#include <string>
#include <stdair/bom/BomAbstract.hpp>
#include <airtsp/bom/OriginDestinationSetKey.hpp>
#include <airtsp/bom/OriginDestinationSetTypes.hpp>
```

### Namespaces

- namespace [boost](#)
- namespace [boost::serialization](#)
- namespace [stdair](#)
- namespace [AIRTSP](#)

## Classes

- class [AIRTSP::OriginDestinationSet](#)  
*Class representing a simple sub-network.*

## 11.25 airtsp/bom/OriginDestinationSetKey.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <boost/archive/text_iarchive.hpp>
#include <boost/archive/text_oarchive.hpp>
#include <boost/serialization/access.hpp>
#include <stdair/basic/BasConst_Inventory.hpp>
#include <airtsp/bom/OriginDestinationSetKey.hpp>
```

## Namespaces

- namespace [AIRTSP](#)

## Functions

- template void [AIRTSP::OriginDestinationSetKey::serialize](#)< [ba::text\\_oarchive](#) > ([ba::text\\_oarchive](#) &, unsigned int)
- template void [AIRTSP::OriginDestinationSetKey::serialize](#)< [ba::text\\_iarchive](#) > ([ba::text\\_iarchive](#) &, unsigned int)

## 11.26 airtsp/bom/OriginDestinationSetKey.hpp File Reference

```
#include <iosfwd>
#include <string>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/bom/KeyAbstract.hpp>
```

## Namespaces

- namespace [boost](#)
- namespace [boost::serialization](#)
- namespace [AIRTSP](#)

## Classes

- struct [AIRTSP::OriginDestinationSetKey](#)  
*Structure representing the key of a sub-network.*



## 11.27 airtsp/bom/OriginDestinationSetTypes.hpp File Reference

```
#include <map>
#include <list>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/bom/key_types.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

### Typedefs

- typedef std::list< OriginDestinationSet \* > [AIRTSP::OriginDestinationSetList\\_T](#)
- typedef std::map< const stdair::MapKey\_T, OriginDestinationSet \* > [AIRTSP::OriginDestinationSetMap\\_T](#)

## 11.28 airtsp/bom/ReachableUniverse.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <boost/archive/text_iarchive.hpp>
#include <boost/archive/text_oarchive.hpp>
#include <boost/serialization/access.hpp>
#include <stdair/basic/BasConst_Inventory.hpp>
#include <airtsp/bom/ReachableUniverse.hpp>
#include <airtsp/bom/SegmentPathPeriod.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

### Functions

- template void [AIRTSP::ReachableUniverse::serialize< ba::text\\_oarchive >](#) (ba::text\_oarchive &, unsigned int)
- template void [AIRTSP::ReachableUniverse::serialize< ba::text\\_iarchive >](#) (ba::text\_iarchive &, unsigned int)

## 11.29 airtsp/bom/ReachableUniverse.hpp File Reference

```
#include <iosfwd>
#include <string>
#include <stdair/bom/BomAbstract.hpp>
```

```
#include <airtsp/bom/ReachableUniverseKey.hpp>
#include <airtsp/bom/ReachableUniverseTypes.hpp>
#include <airtsp/bom/SegmentPathPeriodTypes.hpp>
```

### Namespaces

- namespace [boost](#)
- namespace [boost::serialization](#)
- namespace [stdair](#)
- namespace [AIRTSP](#)

### Classes

- class [AIRTSP::ReachableUniverse](#)  
*Class representing the root of the schedule-related BOM tree.*

## 11.30 airtsp/bom/ReachableUniverseKey.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <boost/archive/text_iarchive.hpp>
#include <boost/archive/text_oarchive.hpp>
#include <boost/serialization/access.hpp>
#include <stdair/basic/BasConst_Inventory.hpp>
#include <airtsp/bom/ReachableUniverseKey.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

### Functions

- template void [AIRTSP::ReachableUniverseKey::serialize< ba::text\\_oarchive >](#) (ba::text\_oarchive &, unsigned int)
- template void [AIRTSP::ReachableUniverseKey::serialize< ba::text\\_iarchive >](#) (ba::text\_iarchive &, unsigned int)

## 11.31 airtsp/bom/ReachableUniverseKey.hpp File Reference

```
#include <iosfwd>
#include <string>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/bom/KeyAbstract.hpp>
```

### Namespaces

- namespace [boost](#)
- namespace [boost::serialization](#)
- namespace [AIRTSP](#)

### Classes

- struct [AIRTSP::ReachableUniverseKey](#)  
*Structure representing the key of the schedule-related BOM tree root.*

## 11.32 airtsp/bom/ReachableUniverseTypes.hpp File Reference

```
#include <map>
#include <list>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/bom/key_types.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

### Typedefs

- typedef std::list< ReachableUniverse \* > [AIRTSP::ReachableUniverseList\\_T](#)
- typedef std::map< const stdair::MapKey\_T, ReachableUniverse \* > [AIRTSP::ReachableUniverse-Map\\_T](#)

## 11.33 airtsp/bom/SegmentCabinStruct.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/bom/SegmentCabin.hpp>
#include <airtsp/bom/SegmentCabinStruct.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

## 11.34 airtsp/bom/SegmentCabinStruct.hpp File Reference

```
#include <string>
#include <vector>
#include <stdair/stdair_inventory_types.hpp>
```

```
#include <stdair/basic/StructAbstract.hpp>
#include <airtsp/bom/FareFamilyStruct.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [AIRTSP](#)

### Classes

- struct [AIRTSP::SegmentCabinStruct](#)

### Typedefs

- typedef std::vector< SegmentCabinStruct > [AIRTSP::SegmentCabinStructList\\_T](#)

## 11.35 airtsp/bom/SegmentPathPeriod.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <boost/archive/text_iarchive.hpp>
#include <boost/archive/text_oarchive.hpp>
#include <boost/serialization/access.hpp>
#include <stdair/basic/BasConst_General.hpp>
#include <stdair/basic/BasConst_Inventory.hpp>
#include <stdair/basic/BasConst_Period_BOM.hpp>
#include <stdair/basic/BasConst_TravelSolution.hpp>
#include <stdair/bom/Inventory.hpp>
#include <stdair/bom/FlightPeriod.hpp>
#include <stdair/bom/SegmentPeriod.hpp>
#include <stdair/bom/BomManager.hpp>
#include <airtsp/bom/SegmentPathPeriod.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

### Functions

- template void [AIRTSP::SegmentPathPeriod::serialize< ba::text\\_oarchive >](#) (ba::text\_oarchive &, unsigned int)
- template void [AIRTSP::SegmentPathPeriod::serialize< ba::text\\_iarchive >](#) (ba::text\_iarchive &, unsigned int)

## 11.36 airtsp/bom/SegmentPathPeriod.hpp File Reference

```
#include <iosfwd>
#include <string>
#include <stdair/bom/BomAbstract.hpp>
#include <airtsp/bom/SegmentPathPeriodKey.hpp>
#include <airtsp/bom/SegmentPathPeriodTypes.hpp>
```

### Namespaces

- namespace [boost](#)
- namespace [boost::serialization](#)
- namespace [stdair](#)
- namespace [AIRTSP](#)

### Classes

- class [AIRTSP::SegmentPathPeriod](#)  
*Class representing a segment/path.*

## 11.37 airtsp/bom/SegmentPathPeriodKey.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <boost/archive/text_iarchive.hpp>
#include <boost/archive/text_oarchive.hpp>
#include <boost/serialization/access.hpp>
#include <stdair/basic/BasConst_General.hpp>
#include <stdair/basic/BasConst_Inventory.hpp>
#include <stdair/basic/BasConst_Period_BOM.hpp>
#include <stdair/basic/BasConst_TravelSolution.hpp>
#include <airtsp/bom/SegmentPathPeriodKey.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

### Functions

- template void [AIRTSP::SegmentPathPeriodKey::serialize< ba::text\\_oarchive >](#) (ba::text\_oarchive &, unsigned int)
- template void [AIRTSP::SegmentPathPeriodKey::serialize< ba::text\\_iarchive >](#) (ba::text\_iarchive &, unsigned int)

## 11.38 airtsp/bom/SegmentPathPeriodKey.hpp File Reference

```
#include <iosfwd>
#include <string>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_date_time_types.hpp>
#include <stdair/bom/KeyAbstract.hpp>
#include <stdair/bom/PeriodStruct.hpp>
#include <airtsp/bom/SegmentPathPeriodTypes.hpp>
```

### Namespaces

- namespace [boost](#)
- namespace [boost::serialization](#)
- namespace [AIRTSP](#)

### Classes

- struct [AIRTSP::SegmentPathPeriodKey](#)  
*Structure representing the key of a segment/path.*

## 11.39 airtsp/bom/SegmentPathPeriodTypes.hpp File Reference

```
#include <map>
#include <vector>
#include <list>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_date_time_types.hpp>
#include <stdair/bom/key_types.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

### Typedefs

- typedef std::list< SegmentPathPeriod \* > [AIRTSP::SegmentPathPeriodList\\_T](#)
- typedef std::multimap< const stdair::MapKey\_T, SegmentPathPeriod \* > [AIRTSP::SegmentPathPeriodMultimap\\_T](#)
- typedef std::vector< const SegmentPathPeriod \* > [AIRTSP::SegmentPathPeriodLightList\\_T](#)
- typedef std::vector< [SegmentPathPeriodLightList\\_T](#) > [AIRTSP::SegmentPathPeriodListList\\_T](#)
- typedef std::vector< stdair::DateOffset\_T > [AIRTSP::DateOffsetList\\_T](#)

## 11.40 airtsp/bom/SegmentPeriodHelper.cpp File Reference

```
#include <cassert>
#include <stdair/basic/BasConst_General.hpp>
#include <stdair/bom/SegmentPeriod.hpp>
#include <airtsp/bom/SegmentPeriodHelper.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

## 11.41 airtsp/bom/SegmentPeriodHelper.hpp File Reference

```
#include <airtsp/bom/LegStruct.hpp>
#include <airtsp/bom/SegmentStruct.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [AIRTSP](#)

### Classes

- class [AIRTSP::SegmentPeriodHelper](#)  
*Class representing the actual business functions for an airline segment-period.*

## 11.42 airtsp/bom/SegmentStruct.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/bom/SegmentDate.hpp>
#include <airtsp/bom/SegmentStruct.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

## 11.43 airtsp/bom/SegmentStruct.hpp File Reference

```
#include <string>
#include <vector>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
```

```
#include <airtsp/bom/SegmentCabinStruct.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [AIRTSP](#)

### Classes

- struct [AIRTSP::SegmentStruct](#)

### Typedefs

- typedef std::vector< SegmentStruct > [AIRTSP::SegmentStructList\\_T](#)

## 11.44 airtsp/command/InventoryGenerator.cpp File Reference

```
#include <cassert>
#include <boost/date_time/date_iterator.hpp>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/basic/BasConst_Inventory.hpp>
#include <stdair/bom/BomManager.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/bom/Inventory.hpp>
#include <stdair/bom/AirlineFeature.hpp>
#include <stdair/bom/FlightPeriod.hpp>
#include <stdair/bom/SegmentPeriod.hpp>
#include <stdair/factory/FacBomManager.hpp>
#include <stdair/service/Logger.hpp>
#include <airtsp/bom/FlightPeriodStruct.hpp>
#include <airtsp/bom/SegmentPeriodHelper.hpp>
#include <airtsp/command/InventoryGenerator.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

## 11.45 airtsp/command/InventoryGenerator.hpp File Reference

```
#include <stdair/command/CmdAbstract.hpp>
#include <airtsp/AIRTSP_Types.hpp>
```



### Namespaces

- namespace [stdair](#)
- namespace [AIRTSP](#)
- namespace [AIRTSP::ScheduleParserHelper](#)

### Classes

- class [AIRTSP::InventoryGenerator](#)

## 11.46 airtsp/command/OnDParser.cpp File Reference

```
#include <cassert>
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/service/Logger.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <airtsp/command/OnDParserHelper.hpp>
#include <airtsp/command/OnDParser.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

## 11.47 airtsp/command/OnDParser.hpp File Reference

```
#include <string>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_file.hpp>
#include <stdair/command/CmdAbstract.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [AIRTSP](#)

### Classes

- class [AIRTSP::OnDParser](#)  
*Class wrapping the parser entry point.*

## 11.48 airtsp/command/OnDParserHelper.cpp File Reference

```
#include <cassert>
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/service/Logger.hpp>
#include <airtsp/command/OnDParserHelper.hpp>
#include <airtsp/command/OnDPeriodGenerator.hpp>
```

### Namespaces

- namespace [AIRTSP](#)
- namespace [AIRTSP::OnDParserHelper](#)

### Functions

- [chset\\_t AIRTSP::OnDParserHelper::alpha\\_cap\\_set\\_p](#) ("A-Z")
- [repeat\\_p\\_t AIRTSP::OnDParserHelper::airport\\_p](#) ([chset\\_t](#)("0-9A-Z").derived(), 3, 3)
- [repeat\\_p\\_t AIRTSP::OnDParserHelper::airline\\_code\\_p](#) ([alpha\\_cap\\_set\\_p](#).derived(), 2, 3)
- [bounded4\\_p\\_t AIRTSP::OnDParserHelper::year\\_p](#) ([uint4\\_p](#).derived(), 2000u, 2099u)
- [bounded2\\_p\\_t AIRTSP::OnDParserHelper::month\\_p](#) ([uint2\\_p](#).derived(), 1u, 12u)
- [bounded2\\_p\\_t AIRTSP::OnDParserHelper::day\\_p](#) ([uint2\\_p](#).derived(), 1u, 31u)
- [bounded2\\_p\\_t AIRTSP::OnDParserHelper::hours\\_p](#) ([uint2\\_p](#).derived(), 0u, 23u)
- [bounded2\\_p\\_t AIRTSP::OnDParserHelper::minutes\\_p](#) ([uint2\\_p](#).derived(), 0u, 59u)
- [bounded2\\_p\\_t AIRTSP::OnDParserHelper::seconds\\_p](#) ([uint2\\_p](#).derived(), 0u, 59u)
- [chset\\_t AIRTSP::OnDParserHelper::class\\_code\\_p](#) ("A-Z")

### Variables

- [uint2\\_p\\_t AIRTSP::OnDParserHelper::uint2\\_p](#)
- [uint4\\_p\\_t AIRTSP::OnDParserHelper::uint4\\_p](#)
- [uint1\\_4\\_p\\_t AIRTSP::OnDParserHelper::uint1\\_4\\_p](#)

## 11.49 airtsp/command/OnDParserHelper.hpp File Reference

```
#include <string>
#include <boost/date_time/posix_time/posix_time.hpp>
#include <boost/date_time/gregorian/gregorian.hpp>
#include <stdair/command/CmdAbstract.hpp>
#include <airtsp/AIRTSP_Types.hpp>
#include <airtsp/basic/BasParserTypes.hpp>
#include <airtsp/bom/OnDPeriodStruct.hpp>
```

## Namespaces

- namespace [stdair](#)
- namespace [AIRTSP](#)
- namespace [AIRTSP::OnDParserHelper](#)

## Classes

- struct [AIRTSP::OnDParserHelper::ParserSemanticAction](#)
- struct [AIRTSP::OnDParserHelper::storeOrigin](#)
- struct [AIRTSP::OnDParserHelper::storeDestination](#)
- struct [AIRTSP::OnDParserHelper::storeDateRangeStart](#)
- struct [AIRTSP::OnDParserHelper::storeDateRangeEnd](#)
- struct [AIRTSP::OnDParserHelper::storeStartRangeTime](#)
- struct [AIRTSP::OnDParserHelper::storeEndRangeTime](#)
- struct [AIRTSP::OnDParserHelper::storeAirlineCode](#)
- struct [AIRTSP::OnDParserHelper::storeClassCode](#)
- struct [AIRTSP::OnDParserHelper::doEndOnD](#)
- struct [AIRTSP::OnDParserHelper::OnDParser](#)
- struct [AIRTSP::OnDParserHelper::OnDParser::definition< ScannerT >](#)
- class [AIRTSP::OnDPeriodFileParser](#)

## 11.50 airtsp/command/OnDPeriodGenerator.cpp File Reference

```
#include <cassert>
#include <stdair/stdair_date_time_types.hpp>
#include <stdair/bom/BomManager.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/factory/FacBomManager.hpp>
#include <stdair/service/Logger.hpp>
#include <airtsp/bom/OnDPeriodStruct.hpp>
#include <airtsp/command/OnDPeriodGenerator.hpp>
```

## Namespaces

- namespace [AIRTSP](#)

## 11.51 airtsp/command/OnDPeriodGenerator.hpp File Reference

```
#include <stdair/command/CmdAbstract.hpp>
#include <airtsp/AIRTSP_Types.hpp>
```

## Namespaces

- namespace [stdair](#)
- namespace [AIRTSP](#)
- namespace [AIRTSP::OnDParserHelper](#)

## Classes

- class [AIRTSP::OnDPeriodGenerator](#)

*Class handling the generation / instantiation of the O&D-Period BOM.*

## 11.52 airtsp/command/ScheduleParser.cpp File Reference

```
#include <cassert>
#include <string>
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/service/Logger.hpp>
#include <airtsp/command/SegmentPathGenerator.hpp>
#include <airtsp/command/ScheduleParserHelper.hpp>
#include <airtsp/command/ScheduleParser.hpp>
```

## Namespaces

- namespace [AIRTSP](#)

## 11.53 airtsp/command/ScheduleParser.hpp File Reference

```
#include <string>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/command/CmdAbstract.hpp>
#include <stdair/stdair_file.hpp>
```

## Namespaces

- namespace [stdair](#)
- namespace [AIRTSP](#)

## Classes

- class [AIRTSP::ScheduleParser](#)

## 11.54 airtsp/command/ScheduleParserHelper.cpp File Reference

```
#include <cassert>
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/service/Logger.hpp>
```

```
#include <airtsp/command/ScheduleParserHelper.hpp>
#include <airtsp/command/InventoryGenerator.hpp>
```

## Namespaces

- namespace [AIRTSP](#)
- namespace [AIRTSP::ScheduleParserHelper](#)

## Functions

- [repeat\\_p\\_t AIRTSP::ScheduleParserHelper::airline\\_code\\_p](#) ([chset\\_t](#)("0-9A-Z").derived(), 2, 3)
- [bounded1\\_4\\_p\\_t AIRTSP::ScheduleParserHelper::flight\\_number\\_p](#) ([uint1\\_4\\_p](#).derived(), 0u, 9999u)
- [bounded4\\_p\\_t AIRTSP::ScheduleParserHelper::year\\_p](#) ([uint4\\_p](#).derived(), 2000u, 2099u)
- [bounded2\\_p\\_t AIRTSP::ScheduleParserHelper::month\\_p](#) ([uint2\\_p](#).derived(), 1u, 12u)
- [bounded2\\_p\\_t AIRTSP::ScheduleParserHelper::day\\_p](#) ([uint2\\_p](#).derived(), 1u, 31u)
- [repeat\\_p\\_t AIRTSP::ScheduleParserHelper::dow\\_p](#) ([chset\\_t](#)("0-1").derived().derived(), 7, 7)
- [repeat\\_p\\_t AIRTSP::ScheduleParserHelper::airport\\_p](#) ([chset\\_t](#)("0-9A-Z").derived(), 3, 3)
- [bounded2\\_p\\_t AIRTSP::ScheduleParserHelper::hours\\_p](#) ([uint2\\_p](#).derived(), 0u, 23u)
- [bounded2\\_p\\_t AIRTSP::ScheduleParserHelper::minutes\\_p](#) ([uint2\\_p](#).derived(), 0u, 59u)
- [bounded2\\_p\\_t AIRTSP::ScheduleParserHelper::seconds\\_p](#) ([uint2\\_p](#).derived(), 0u, 59u)
- [chset\\_t AIRTSP::ScheduleParserHelper::cabin\\_code\\_p](#) ("A-Z")
- [repeat\\_p\\_t AIRTSP::ScheduleParserHelper::key\\_p](#) ([chset\\_t](#)("0-9A-Z").derived(), 1, 10)
- [repeat\\_p\\_t AIRTSP::ScheduleParserHelper::class\\_code\\_list\\_p](#) ([chset\\_t](#)("A-Z").derived(), 1, 26)

## Variables

- [int1\\_p\\_t AIRTSP::ScheduleParserHelper::int1\\_p](#)
- [uint2\\_p\\_t AIRTSP::ScheduleParserHelper::uint2\\_p](#)
- [uint4\\_p\\_t AIRTSP::ScheduleParserHelper::uint4\\_p](#)
- [uint1\\_4\\_p\\_t AIRTSP::ScheduleParserHelper::uint1\\_4\\_p](#)
- [int1\\_p\\_t AIRTSP::ScheduleParserHelper::family\\_code\\_p](#)

## 11.55 airtsp/command/ScheduleParserHelper.hpp File Reference

```
#include <string>
#include <stdair/command/CmdAbstract.hpp>
#include <airtsp/AIRTSP_Types.hpp>
#include <airtsp/basic/BasParserTypes.hpp>
#include <airtsp/bom/FlightPeriodStruct.hpp>
```

## Namespaces

- namespace [stdair](#)
- namespace [AIRTSP](#)
- namespace [AIRTSP::ScheduleParserHelper](#)

## Classes

- struct [AIRTSP::ScheduleParserHelper::ParserSemanticAction](#)
- struct [AIRTSP::ScheduleParserHelper::storeAirlineCode](#)
- struct [AIRTSP::ScheduleParserHelper::storeFlightNumber](#)
- struct [AIRTSP::ScheduleParserHelper::storeDateRangeStart](#)
- struct [AIRTSP::ScheduleParserHelper::storeDateRangeEnd](#)
- struct [AIRTSP::ScheduleParserHelper::storeDow](#)
- struct [AIRTSP::ScheduleParserHelper::storeLegBoardingPoint](#)
- struct [AIRTSP::ScheduleParserHelper::storeLegOffPoint](#)
- struct [AIRTSP::ScheduleParserHelper::storeOperatingAirlineCode](#)
- struct [AIRTSP::ScheduleParserHelper::storeOperatingFlightNumber](#)
- struct [AIRTSP::ScheduleParserHelper::storeBoardingTime](#)
- struct [AIRTSP::ScheduleParserHelper::storeOffTime](#)
- struct [AIRTSP::ScheduleParserHelper::storeElapsedTime](#)
- struct [AIRTSP::ScheduleParserHelper::storeLegCabinCode](#)
- struct [AIRTSP::ScheduleParserHelper::storeCapacity](#)
- struct [AIRTSP::ScheduleParserHelper::storeSegmentSpecificity](#)
- struct [AIRTSP::ScheduleParserHelper::storeSegmentBoardingPoint](#)
- struct [AIRTSP::ScheduleParserHelper::storeSegmentOffPoint](#)
- struct [AIRTSP::ScheduleParserHelper::storeSegmentCabinCode](#)
- struct [AIRTSP::ScheduleParserHelper::storeClasses](#)
- struct [AIRTSP::ScheduleParserHelper::storeFamilyCode](#)
- struct [AIRTSP::ScheduleParserHelper::storeFRAT5CurveKey](#)
- struct [AIRTSP::ScheduleParserHelper::storeFFDisutilityCurveKey](#)
- struct [AIRTSP::ScheduleParserHelper::storeFCClasses](#)
- struct [AIRTSP::ScheduleParserHelper::doEndFlight](#)
- struct [AIRTSP::ScheduleParserHelper::FlightPeriodParser](#)
- struct [AIRTSP::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >](#)
- class [AIRTSP::FlightPeriodFileParser](#)

## 11.56 airtsp/command/SegmentPathGenerator.cpp File Reference

```
#include <cassert>
#include <vector>
#include <stdair/basic/BasConst_Inventory.hpp>
#include <stdair/bom/BomManager.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/bom/Inventory.hpp>
#include <stdair/bom/FlightPeriod.hpp>
#include <stdair/bom/SegmentPeriod.hpp>
#include <stdair/factory/FacBomManager.hpp>
#include <stdair/service/Logger.hpp>
#include <airtsp/bom/ReachableUniverse.hpp>
#include <airtsp/bom/OriginDestinationSet.hpp>
#include <airtsp/bom/SegmentPathPeriod.hpp>
#include <airtsp/command/SegmentPathGenerator.hpp>
```

## Namespaces

- namespace [AIRTSP](#)

## 11.57 airtsp/command/SegmentPathGenerator.hpp File Reference

```
#include <vector>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/command/CmdAbstract.hpp>
#include <airtsp/AIRTSP_Types.hpp>
```

## Namespaces

- namespace [stdair](#)
- namespace [AIRTSP](#)

## Classes

- class [AIRTSP::SegmentPathGenerator](#)  
*Class handling the generation / instantiation of the network BOM.*

## 11.58 airtsp/command/SegmentPathProvider.cpp File Reference

```
#include <cassert>
#include <string>
#include <sstream>
#include <stdair/basic/BasConst_BomDisplay.hpp>
#include <stdair/bom/BomManager.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/bom/Inventory.hpp>
#include <stdair/bom/FlightPeriod.hpp>
#include <stdair/bom/SegmentPeriod.hpp>
#include <stdair/bom/BookingRequestStruct.hpp>
#include <stdair/bom/TravelSolutionStruct.hpp>
#include <stdair/service/Logger.hpp>
#include <airtsp/basic/BasConst_General.hpp>
#include <airtsp/bom/ReachableUniverse.hpp>
#include <airtsp/bom/OriginDestinationSet.hpp>
#include <airtsp/bom/SegmentPathPeriod.hpp>
#include <airtsp/command/SegmentPathProvider.hpp>
```

## Namespaces

- namespace [AIRTSP](#)

## 11.59 airtsp/command/SegmentPathProvider.hpp File Reference

```
#include <stdair/bom/TravelSolutionTypes.hpp>
#include <stdair/command/CmdAbstract.hpp>
```

## Namespaces

- namespace [stdair](#)
- namespace [AIRTSP](#)

## Classes

- class [AIRTSP::SegmentPathProvider](#)  
*Class building the travel solutions from airline schedules.*

## 11.60 airtsp/command/Simulator.cpp File Reference

```
#include <cassert>
#include <string>
#include <sstream>
#include <stdair/basic/BasConst_General.hpp>
#include <stdair/bom/BomManager.hpp>
#include <stdair/bom/BookingRequestStruct.hpp>
#include <stdair/service/Logger.hpp>
#include <airtsp/command/Simulator.hpp>
```

## Namespaces

- namespace [AIRTSP](#)

## 11.61 airtsp/command/Simulator.hpp File Reference

```
#include <stdair/command/CmdAbstract.hpp>
```

## Namespaces

- namespace [stdair](#)
- namespace [AIRTSP](#)



## Classes

- class [AIRTSP::Simulator](#)

## 11.62 airtsp/command/TravelSolutionParser.cpp File Reference

```
#include <sstream>
#include <fstream>
#include <cassert>
#include <stdair/stdair_exceptions.hpp>
#include <stdair/basic/BasConst_TravelSolution.hpp>
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/service/Logger.hpp>
#include <airtsp/command/TravelSolutionParser.hpp>
```

## Namespaces

- namespace [AIRTSP](#)

## 11.63 airtsp/command/TravelSolutionParser.hpp File Reference

```
#include <string>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/command/CmdAbstract.hpp>
```

## Namespaces

- namespace [AIRTSP](#)

## Classes

- class [AIRTSP::TravelSolutionParser](#)

*Class filling the TravelSolutionHolder structure (representing a list of classes/travelSolutions) from a given input file.*

## 11.64 airtsp/config/airtsp-paths.hpp.in File Reference

## 11.65 airtsp/factory/FacAIRTSPServiceContext.cpp File Reference

```
#include <cassert>
#include <stdair/service/FacSupervisor.hpp>
```

```
#include <airtsp/factory/FacAIRTSPServiceContext.hpp>
#include <airtsp/service/AIRTSP_ServiceContext.hpp>
```

#### Namespaces

- namespace [AIRTSP](#)

### 11.66 airtsp/factory/FacAIRTSPServiceContext.hpp File Reference

```
#include <stdair/stdair_basic_types.hpp>
#include <stdair/service/FacServiceAbstract.hpp>
```

#### Namespaces

- namespace [AIRTSP](#)

#### Classes

- class [AIRTSP::FacAIRTSPServiceContext](#)  
*Factory for the service context.*

### 11.67 airtsp/factory/FacServiceAbstract.cpp File Reference

```
#include <cassert>
#include <airtsp/service/ServiceAbstract.hpp>
#include <airtsp/factory/FacServiceAbstract.hpp>
```

#### Namespaces

- namespace [AIRTSP](#)

### 11.68 airtsp/factory/FacServiceAbstract.hpp File Reference

```
#include <vector>
```

#### Namespaces

- namespace [AIRTSP](#)

#### Classes

- class [AIRTSP::FacServiceAbstract](#)

## 11.69 airtsp/service/AIRTSP\_Service.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <boost/make_shared.hpp>
#include <stdair/basic/BasChronometer.hpp>
#include <stdair/bom/BomManager.hpp>
#include <stdair/bom/BookingRequestStruct.hpp>
#include <stdair/bom/TravelSolutionStruct.hpp>
#include <stdair/service/Logger.hpp>
#include <stdair/STDAIR_Service.hpp>
#include <airtsp/basic/BasConst_AIRTSP_Service.hpp>
#include <airtsp/factory/FacAIRTSPServiceContext.hpp>
#include <airtsp/command/Simulator.hpp>
#include <airtsp/command/ScheduleParser.hpp>
#include <airtsp/command/OnDParser.hpp>
#include <airtsp/command/SegmentPathProvider.hpp>
#include <airtsp/command/InventoryGenerator.hpp>
#include <airtsp/command/SegmentPathGenerator.hpp>
#include <airtsp/service/AIRTSP_ServiceContext.hpp>
#include <airtsp/AIRTSP_Service.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

## 11.70 airtsp/service/AIRTSP\_ServiceContext.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/STDAIR_Service.hpp>
#include <airtsp/basic/BasConst_AIRTSP_Service.hpp>
#include <airtsp/service/AIRTSP_ServiceContext.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

## 11.71 airtsp/service/AIRTSP\_ServiceContext.hpp File Reference

```
#include <string>
```

```
#include <boost/shared_ptr.hpp>
#include <stdair/stdair_service_types.hpp>
#include <stdair/service/ServiceAbstract.hpp>
#include <airtsp/AIRTSP_Types.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

### Classes

- class [AIRTSP::AIRTSP\\_ServiceContext](#)  
*Class holding the context of the Airtsp services.*

## 11.72 airtsp/service/ServiceAbstract.cpp File Reference

```
#include <airtsp/service/ServiceAbstract.hpp>
```

### Namespaces

- namespace [AIRTSP](#)

## 11.73 airtsp/service/ServiceAbstract.hpp File Reference

```
#include <iostream>
#include <sstream>
```

### Namespaces

- namespace [AIRTSP](#)

### Classes

- class [AIRTSP::ServiceAbstract](#)

### Functions

- `template<class charT, class traits> std::basic_ostream< charT, traits > & operator<< (std::basic_ostream< charT, traits > &ioOut, const AIRTSP::ServiceAbstract &iService)`
- `template<class charT, class traits> std::basic_istream< charT, traits > & operator>> (std::basic_istream< charT, traits > &ioIn, AIRTSP::ServiceAbstract &ioService)`

### 11.73.1 Function Documentation

**11.73.1.1** `template<class charT, class traits> std::basic_ostream<charT, traits>& operator<< (std::basic_ostream< charT, traits > & ioOut, const AIRTSP::ServiceAbstract & iService) [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 42 of file ServiceAbstract.hpp.

**11.73.1.2** `template<class charT, class traits> std::basic_istream<charT, traits>& operator>> (std::basic_istream< charT, traits > & ioIn, AIRTSP::ServiceAbstract & ioService) [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 70 of file ServiceAbstract.hpp.

References `AIRTSP::ServiceAbstract::fromStream()`.

**11.74 doc/local/authors.doc File Reference**

**11.75 doc/local/codingrules.doc File Reference**

**11.76 doc/local/copyright.doc File Reference**

**11.77 doc/local/documentation.doc File Reference**

**11.78 doc/local/features.doc File Reference**

**11.79 doc/local/help\_wanted.doc File Reference**

**11.80 doc/local/howto\_release.doc File Reference**

**11.81 doc/local/index.doc File Reference**

**11.82 doc/local/installation.doc File Reference**

**11.83 doc/local/linking.doc File Reference**

**11.84 doc/local/test.doc File Reference**

**11.85 doc/local/users\_guide.doc File Reference**

**11.86 doc/local/verification.doc File Reference**

**11.87 doc/tutorial/tutorial.doc File Reference**

**11.88 test/airtsp/AirlineScheduleTestSuite.cpp File Reference**

**11.89 test/airtsp/AirlineScheduleTestSuite.hpp File Reference**

```
#include <sstream>
```

```
#include <cppunit/extensions/HelperMacros.h>
```

#### Classes

- class [AirlineScheduleTestSuite](#)

#### Functions

- [CPPUNIT\\_TEST\\_SUITE\\_REGISTRATION](#) ([AirlineScheduleTestSuite](#))

#### 11.89.1 Function Documentation

**11.89.1.1 CPPUNIT\_TEST\_SUITE\_REGISTRATION ([AirlineScheduleTestSuite](#))**

## 12 AirTSP Page Documentation

### 12.1 Configuration helper for AirTSP programs

```

*/
#ifndef __AIRTSP_PATHS_HPP__
#define __AIRTSP_PATHS_HPP__

#define PACKAGE "@PACKAGE@"
#define PACKAGE_NAME "@PACKAGE_NAME@"
#define PACKAGE_VERSION "@PACKAGE_VERSION@"
#define PREFIXDIR "@prefix@"
#define EXEC_PREFIX "@exec_prefix@"
#define BINDIR "@bindir@"
#define LIBDIR "@libdir@"
#define LIBEXECDIR "@libexecdir@"
#define SBINDIR "@sbindir@"
#define SYSCONFDIR "@sysconfdir@"
#define INCLUDEDIR "@includedir@"
#define DATAROOTDIR "@datarootdir@"
#define DATADIR "@datadir@"
#define DOCDIR "@docdir@"
#define MANDIR "@mandir@"
#define INFODIR "@infodir@"
#define HTMLDIR "@htmldir@"
#define PDFDIR "@pdfdir@"
#define STDAIR_SAMPLE_DIR "@sampledir@"

#endif // __AIRTSP_PATHS_HPP__

/*!
```

### 12.2 People

#### 12.2.1 Project Admins

- Denis Arnaud <denis\_arnaud@users.sourceforge.net> ([N](#))
- Anh Quan Nguyen <quannaus@users.sourceforge.net> ([N](#))

#### 12.2.2 Developers

- Anh Quan Nguyen <quannaus@users.sourceforge.net> ([N](#))
- Denis Arnaud <denis\_arnaud@users.sourceforge.net> ([N](#))
- Gabrielle Sabatier <gsabatier@users.sourceforge.net> ([N](#))

#### 12.2.3 Retired Developers

- Daniel Perez <daniperez@users.sourceforge.net> ([N](#))
- Mehdi Ayouni <mehdi.ayouni@gmail.com>
- Son Nguyen Kim <snguyenkim@users.sourceforge.net>
- Alexandre Point <apoint@users.sourceforge.net>

### 12.2.4 Contributors

- Emmanuel Bastien <ebastien@users.sourceforge.net> ([N](#))
- Christophe Lacombe <ddtof@users.sourceforge.net> ([N](#))

### 12.2.5 Distribution Maintainers

- [Fedora/RedHat](#): Denis Arnaud <denis\_arnaud@users.sourceforge.net> ([N](#))
- [Debian](#): Emmanuel Bastien <ebastien@users.sourceforge.net> ([N](#))

**Note:**

(N) - [Amadeus](#) employees.

## 12.3 Coding Rules

In the following sections we describe the naming conventions which are used for files, classes, structures, local variables, and global variables.

### 12.3.1 Default Naming Rules for Variables

Variables names follow Java naming conventions. Examples:

- `lNumberOfPassengers`
- `lSeatAvailability`

### 12.3.2 Default Naming Rules for Functions

Function names follow Java naming conventions. Example:

- `int myFunctionName (const int& a, int b)`

### 12.3.3 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`

### 12.3.4 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`



### 12.3.5 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.

## 12.4 Copyright and License

### 12.4.1 GNU LESSER GENERAL PUBLIC LICENSE

#### 12.4.1.1 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.]

#### 12.4.2 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.

### 12.4.3 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.

**12.4.3.1 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.

#### **12.4.3.2 END OF TERMS AND CONDITIONS**

### 12.4.4 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

## 12.5 Documentation Rules

### 12.5.1 General Rules

All classes in AirTSP 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 AirTSP 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
};

```

### 12.5.2 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.
 *
 * -----
 *
 * AirTSP - C++ Airline Travel Solution Provider Library
 *
 * Copyright (C) 2009-2010 (\see authors file for a list of contributors)
 *
 * \see copyright file for license information
 *
 * -----
 */

```



### 12.5.3 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);
```

## 12.6 Main features

A short list of the main features of AirTSP is given below sorted in different categories. Many more features and functions exist and for these we refer to the reference documentation.

### 12.6.1 Network generation

- Network/graph generation

### 12.6.2 Finding travel solutions

- Matching of travel solutions with user requests

### 12.6.3 Other features

- CSV input file parsing
- Memory handling

## 12.7 Make a Difference

**Do not ask what AirTSP can do for you. Ask what you can do for AirTSP.**

You can help us to develop the AirTSP library. There are always a lot of things you can do:

- Start using AirTSP

- Tell your friends about AirTSP 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 AirTSP discussion forums on SourceForge. If you know the answer to a question, help others to overcome their AirTSP 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 AirTSP to new platforms. If you manage to compile AirTSP on a new platform, then tell us how you did it.
- Send us your code. If you have a good AirTSP compatible code, which you can release under the LGPLv2.1, and you think it should be included in AirTSP, then send it to us.
- Become an AirTSP developer. Send us an e-mail and tell what you can do for AirTSP.

## 12.8 Make a new release

### 12.8.1 Introduction

This document describes briefly the recommended procedure of releasing a new version of AirTSP using a Linux development machine and the SourceForge project site.

The following steps are required to make a release of the distribution package.

### 12.8.2 Initialisation

Clone locally the full [Git project](#):

```
cd ~
mkdir -p dev/sim
cd ~/dev/sim
git clone git://airtsp.git.sourceforge.net/gitroot/airtsp/airtsp airtspgit
cd airtspgit
git checkout trunk
```

### 12.8.3 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/airtspgit
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 airtsp.spec
```

### 12.8.4 Commit and publish the release branch

Commit the new release:

```
cd ~/dev/sim/airtspgit
git add -A
git commit -m "[Release 0.5.0] Release of the 0.5.0 version of AirTSP."
git push
```

### 12.8.5 Create distribution packages

Create the distribution packages using the following command:

```
cd ~/dev/sim/airtspgit
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}/airtsp-0.5.0 \
      -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, airtsp-0.5.0.tar.gz and airtsp-0.5.0.tar.bz2.

### 12.8.6 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/airtspgit/build
git checkout releases
rsync -aiv ${INSTALL_BASEDIR}/airtsp-0.5.0/share/doc/airtsp-0.5.0/html/ \
  your_sf_user,airtsp@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**.

### 12.8.7 Generate the RPM packages

Optionally, generate the RPM package (for instance, for [Fedora/RedHat](#)):

```
cd ~/dev/sim/airtspgit/build
git checkout releases
make dist
```

To perform this step, rpm-build, rpmlint and rpmdevtools have to be available on the system.

```
cp ../airtsp.spec ~/dev/packages/SPECS \
  && cp airtsp-0.5.0.tar.bz2 ~/dev/packages/SOURCES
cd ~/dev/packages/SPECS
rpmbuild -ba airtsp.spec
cd ~/dev/packages
rpmlint -i SPECS/airtsp.spec SRPMS/airtsp-0.5.0-1.fc16.src.rpm \
  RPMS/noarch/airtsp-* RPMS/i686/airtsp-*
```

### 12.8.8 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 [AirTSP's Git repository](#).

### 12.8.9 Create the binary package, including the documentation

Create the binary package, which includes HTML and PDF documentation, using the following command:

```
cd ~/dev/sim/airtspgit/build
git checkout releases
make package
```

The output binary package will be named, for instance, `airtsp-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.

### 12.8.10 Upload the files to SourceForge

Upload the distribution and documentation packages to the SourceForge server. Check [SourceForge help page on uploading software](#).

### 12.8.11 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](#).

### 12.8.12 Send an email on the announcement mailing-list

Finally, you should send an announcement to [airtsp-announce@lists.sourceforge.net](mailto:airtsp-announce@lists.sourceforge.net) (see <https://lists.sourceforge.net/lists/listinfo/airtsp-announce> for the archives)

## 12.9 Installation

### 12.9.1 Table of Contents

- [Fedora/RedHat Linux distributions](#)
- [AirTSP 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](#)

### 12.9.2 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 airtsp-devel airtsp-doc
```

RPM packages can also be available on the [SourceForge download site](#).

### 12.9.3 AirTSP Requirements

AirTSP 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 AirTSP.

#### 12.9.4 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 a file ``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.

### 12.9.5 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.

### 12.9.6 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.

### 12.9.7 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.

### 12.9.8 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'`.

### 12.9.9 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
```

### 12.9.10 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'`.

### 12.9.11 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.

### 12.9.12 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
```

### 12.9.13 'cmake' Invocation

'cmake' recognizes the following options to control how it operates.

- '-help', '-h' print a summary of all of the options to 'cmake', 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:

```
-- Requires Git without specifying any version
cmake -DCMAKE_INSTALL_PREFIX=/home/user/dev/deliveries/airtsp-99.99.99 -DLIB_SUFFIX=64 -DCMAKE_BUILD_TYPE:
-- 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: 6100bb1479e9c72f807a60067138dfe1b71c7ec7 trunk
-- Requires Boost-1.41
-- Boost version: 1.46.0
-- Found the following Boost libraries:
```

```

-- regex
-- 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: 0.38.0
-- 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 'airtsplib' to CXX
-- Test 'AirlineScheduleTestSuite' to be built with 'AirlineScheduleTestSuite.cpp'
--
-- =====
-- -----
-- ---      Project Information      ---
-- -----
-- PROJECT_NAME ..... : airtsp
-- PACKAGE_PRETTY_NAME ..... : AirTSP
-- PACKAGE ..... : airtsp
-- PACKAGE_NAME ..... : AIRTSP
-- PACKAGE_BRIEF ..... : C++ Simulated Airline Schedule Manager Library
-- PACKAGE_VERSION ..... : 99.99.99
-- GENERIC_LIB_VERSION ..... : 99.99.99
-- GENERIC_LIB_SOVERSION ..... : 99.99
--
-- -----
-- ---      Build Configuration      ---
-- -----
-- Modules to build ..... : airtsp
-- Libraries to build/install ..... : airtspplib
-- Binaries to build/install ..... : airtsp
-- Modules to test ..... : airtsp
-- Binaries to test ..... : AirlineScheduleTestSuitetst
--
-- * Module ..... : airtsp
--   + Layers to build ..... : .;basic;bom;factory;command;service
--   + Dependencies on other layers :
--   + Libraries to build/install . : airtspplib
--   + Executables to build/install : airtsp
--   + Tests to perform ..... : AirlineScheduleTestSuitetst
--
-- 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/airtsp/airtspgithub/config/
-- CMAKE_INSTALL_PREFIX ..... : /home/user/dev/deliveries/airtsp-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/airtsp-99.99.99/lib64
-- INSTALL_BIN_DIR ..... : /home/user/dev/deliveries/airtsp-99.99.99/bin
-- INSTALL_INCLUDE_DIR ..... : /home/user/dev/deliveries/airtsp-99.99.99/include
-- INSTALL_DATA_DIR ..... : /home/user/dev/deliveries/airtsp-99.99.99/share
-- INSTALL_SAMPLE_DIR ..... : /home/user/dev/deliveries/airtsp-99.99.99/share/airtsp/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/airtsp/airtspgithub/README
-- CPACK_RESOURCE_FILE_LICENSE .... : /home/user/dev/sim/airtsp/airtspgithub/COPYING
-- CPACK_GENERATOR ..... : TBZ2
-- CPACK_DEBIAN_PACKAGE_DEPENDS ... :
-- CPACK_SOURCE_GENERATOR ..... : TBZ2;TGZ
-- CPACK_SOURCE_PACKAGE_FILE_NAME . : airtsp-99.99.99
--
-- -----
-- --- External libraries ---
-- -----
--
-- * Boost:
--   - Boost_VERSION ..... : 104600
--   - Boost_LIB_VERSION ..... : 1_46
--   - Boost_HUMAN_VERSION ..... : 1.46.0
--   - Boost_INCLUDE_DIRS ..... : /usr/include
--   - Boost required components .. : regex;program_options;date_time;iostreams;serialization;filesystem;u
--   - Boost required libraries ... : optimized;/usr/lib64/libboost_regex-mt.so;debug;/usr/lib64/libboost_
--
-- * 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 ..... : 0.38.0
--   - STDAIR_BINARY_DIRS ..... : /home/user/dev/deliveries/stdair-0.38.0/bin
--   - STDAIR_EXECUTABLES ..... : stdair
--   - STDAIR_LIBRARY_DIRS ..... : /home/user/dev/deliveries/stdair-0.38.0/lib64
--   - STDAIR_LIBRARIES ..... : stdairlib;stdairuiclib
--   - STDAIR_INCLUDE_DIRS ..... : /home/user/dev/deliveries/stdair-0.38.0/include
--   - STDAIR_SAMPLE_DIR ..... : /home/user/dev/deliveries/stdair-0.38.0/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/airtsp/airtspgithub/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_airtsp
[ 96%] Built target airtspplib
[100%] Built target AirlineScheduleTestSuitetst
Scanning dependencies of target check_airsptst
Test project /home/dan/dev/sim/airtsp/airtspgithub/build/test/airtsp
  Start 1: AirlineScheduleTestSuitetst
1/1 Test #1: AirlineScheduleTestSuitetst ..... Passed    0.15 sec

100% tests passed, 0 tests failed out of 1

Total Test time (real) = 0.40 sec
[100%] Built target check_airsptst
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/airtspgit
rm -rf build && mkdir build
cd build
```

to remove everything.

## 12.10 Linking with AirTSP

### 12.10.1 Table of Contents

- [Introduction](#)
- [Dependencies](#)
- [Using the pkg-config command](#)
- [Using the airtsp-config script](#)
- [M4 macro for the GNU Autotools](#)
- [Using AirTSP with dynamic linking](#)

### 12.10.2 Introduction

There are two convenient methods of linking your programs with the AirTSP library. The first one employs the `'pkg-config'` command (see <http://pkgconfig.freedesktop.org/>), whereas the second one uses `'airtsp-config'` script. These methods are shortly described below.

### 12.10.3 Dependencies

The AirTSP library depends on several other C++ components.

**12.10.3.1 StdAir** Among them, as for now, only StdAir has been packaged. The support for StdAir is taken in charge by a dedicated M4 macro file (namely, `'stdair.m4'`), from the configuration script (generated thanks to `'configure.ac'`).



Figure 1: AirTSP Dependencies

### 12.10.4 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 AirTSP based program `'my_prog.cpp'`, you should use the following command:

```
g++ `pkg-config --cflags airtsp` -o my_prog my_prog.cpp `pkg-config --libs airtsp`
```

For more information see the `'pkg-config'` man pages.

### 12.10.5 Using the airtsp-config script

AirTSP provides a shell script called `'airtsp-config'`, which is installed by default in `'$prefix/bin'` (`'/usr/local/bin'`) directory. It can be used to simplify compilation and linking of AirTSP 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++ 'airtsp-config --cflags' -o my_prog_opt my_prog.cpp 'airtsp-config --libs'
```

A list of `'airtsp-config'` options can be obtained by typing:

```
airtsp-config --help
```

If the `'airtsp-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
```

### 12.10.6 M4 macro for the GNU Autotools

A M4 macro file is delivered with AirTSP, namely `'airtsp.m4'`, which can be found in, e.g., `'/usr/share/aclocal'`. When used by a `'configure'` script, thanks to the `'AM_PATH_AirTSP'` macro (specified in the M4 macro file), the following Makefile variables are then defined:

- `'AirTSP_VERSION'` (e.g., defined to 0.23.0)
- `'AirTSP_CFLAGS'` (e.g., defined to `'-I${prefix}/include'`)
- `'AirTSP_LIBS'` (e.g., defined to `'-L${prefix}/lib -lairtsp'`)

### 12.10.7 Using AirTSP with dynamic linking

When using static linking some of the library routines in AirTSP 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 AirTSP library file during your program execution. If you install the AirTSP 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=<AirTSP installation prefix>/lib:$LD_LIBRARY_PATH
```

## 12.11 Test Rules

This section describes rules how the functionality of the IT++ library should be verified. In the `'tests'` subdirectory test files are provided. All functionality should be tested using these test files.



### 12.11.1 The Test File

Each new IT++ module/class should be accompanied with a test file. The test file is an implementation in C++ that tests the functionality of a function/class or a group of functions/classes called modules. The test file should test relevant parameter settings and input/output relations to guarantee correct functionality of the corresponding classes/functions. The test files should be maintained using version control and updated whenever new functionality is added to the IT++ library.

The test 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 file should be placed in the `'tests'` subdirectory and should have a name ending with `'__test.cpp'`.

### 12.11.2 The Reference File

Consider a test file named `'module_test.cpp'`. A reference file named `'module_test.ref'` should accompany the test 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 file.

### 12.11.3 Testing IT++ Library

One can compile and execute all test programs from `'tests'` subdirectory by typing

```
% make check
```

after successful compilation of the IT++ library.

## 12.12 Users Guide

### 12.12.1 Table of Contents

- [Introduction](#)
- [Get Started](#)
  - [Get the AirTSP library](#)
  - [Build the AirTSP project](#)
  - [Build and Run the Tests](#)
  - [Install the AirTSP Project \(Binaries, Documentation\)](#)
- [Input file of AirTSP Project](#)
- [The schedule BOM Tree](#)
  - [Build of the schedule BOM tree](#)
  - [Display of the schedule BOM tree](#)
- [Exploring the Predefined BOM Tree](#)
  - [Airline Network BOM Tree](#)
  - [Airline Schedule BOM Tree](#)

- [Extending the BOM Tree](#)
- [The travel solution calculation procedure](#)

### 12.12.2 Introduction

The `AirTSP` library contains classes for airline business management. This document does not cover all the aspects of the `AirTSP` library. It does however explain the most important things you need to know in order to start using `AirTSP`.

### 12.12.3 Get Started

#### 12.12.3.1 Get the AirTSP library

Clone locally the full [Git project](#):

```
cd ~
mkdir -p dev/sim
cd ~/dev/sim
git clone git://airtsp.git.sourceforge.net/gitroot/airtsp/airtsp airtspgit
cd airtspgit
git checkout trunk
```

#### 12.12.3.2 Build the AirTSP project

Link with `StdAir`, create the distribution package (say, 0.5.0) and compile using the following commands:

```
cd ~/dev/sim/airtspgit
rm -rf build && mkdir -p build
cd build
cmake -DCMAKE_INSTALL_PREFIX=~/dev/deliveries/airtsp-0.5.0 \
      -DWITH_STDAIR_PREFIX=~/dev/deliveries/stdair-stable \
      -DCMAKE_BUILD_TYPE:STRING=Debug -DINSTALL_DOC:BOOL=ON ..
make
```

#### 12.12.3.3 Build and Run the Tests

After building the `AirTSP` project, the following commands run the tests:

```
cd ~/dev/sim/airtspgit
cd build
make check
```

As a result, you should obtain a similar report:

```
[ 0%] Built target hdr_cfg_airtsp
[ 96%] Built target airtspplib
[100%] Built target AirlineScheduleTestSuitetst
Scanning dependencies of target check_airsptst
Test project /home/dan/dev/sim/airtsp/airtspgithub/build/test/airtsp
  Start 1: AirlineScheduleTestSuitetst
1/1 Test #1: AirlineScheduleTestSuitetst ..... Passed    0.15 sec

100% tests passed, 0 tests failed out of 1

Total Test time (real) = 0.40 sec
[100%] Built target check_airsptst
Scanning dependencies of target check
[100%] Built target check
```

**12.12.3.4 Install the AirTSP Project (Binaries, Documentation)** After the step [Build the AirTSP project](#), to install the library and its header files, type:

```
cd ~/dev/sim/airtspgit
cd build
make install
```

You can check that the executables and other required files have been copied into the given final directory:

```
cd ~dev/deliveries/airtsp-0.5.0
```

To generate the AirTSP project documentation, the commands are:

```
cd ~/dev/sim/airtspgit
cd build
make doc
```

The AirTSP project documentation is available in the following formats: HTML, LaTeX. Those documents are available in a subdirectory:

```
cd ~/dev/sim/airtspgit
cd build
cd doc
```

## 12.12.4 Input file of AirTSP Project

The schedule input file structure should look like the following sample:

```
// Flights:   AirlineCode; FlightNumber; Date-Range; ; DOW; Legs; Segments;
// Legs:      BoardPoint; OffPoint; BoardTime; ArrivalDateOffSet; ArrivalTime;
//            ElapsedTime; LegCabins;
// LegCabins: CabinCode; Capacity;
// Segments: Specific;
BA; 9; 2007-04-20; 2007-06-30; 0000011; LHR; BKK; 22:00; 15:15 / +1; 11:15; F; 5; J; 12; W; 20; Y; 300; BK
BA; 9; 2007-04-20; 2007-06-30; 1111100; LHR; BKK; 22:00; 15:15 / +1; 11:15; F; 5; J; 12; W; 20; Y; 300; BK
BA; 117; 2007-04-20; 2007-06-30; 1111111; LHR; JFK; 08:20; 11:00; 07:40; F; 5; J; 12; W; 20; Y; 300; 0; F;
BA; 175; 2007-04-20; 2007-06-30; 1111111; LHR; JFK; 10:55; 13:35; 07:40; F; 5; J; 12; W; 20; Y; 300; 0; F;
BA; 179; 2007-04-20; 2007-06-30; 1111111; LHR; JFK; 18:05; 20:45; 07:40; F; 5; J; 12; W; 20; Y; 300; 0; F;
BA; 207; 2007-04-20; 2007-06-30; 1111111; LHR; MIA; 09:40; 14:25; 09:45; F; 5; J; 12; W; 20; Y; 300; 0; F;
BA; 279; 2007-04-20; 2007-06-30; 1111111; LHR; LAX; 10:05; 13:10; 11:05; F; 5; J; 12; W; 20; Y; 300; 0; F;
```

Each line, beyond the header, represents a schedule entry, i.e., the specification of a given flight-period (see [AIRTSP::FlightPeriodStruct](#)). The fields are as follows:

- Flights section
  - AirlineCode (e.g., BA)
  - FlightNumber (e.g., 9)
  - Start of the flight departure period (e.g., 2007-04-20)
  - End of the flight departure period (e.g., 2007-06-30)
  - Day-Of-the-Week for the flight departure period (DOW) (e.g., 0000011)
  - Leg section
  - Segment section

- Leg section
  - BoardPoint (e.g., LHR)
  - OffPoint (e.g., BKK)
  - BoardTime (e.g., 22:00)
  - ArrivalTime (e.g., 15:15)
  - ArrivalDateOffset (e.g., +1)
  - ElapsedTime (e.g., 11:15)
  - Leg-cabin section
- Leg-cabin section
  - Cabin code (e.g., F, J, W or Y)
  - Capacity (e.g., respectively 5, 12, 20 or 300)
- Segment section
  - Specificity flag:
    - \* 0 means that all the segments behave the same way, i.e., have got the same dressing (distribution and order of the booking classes per cabin)
    - \* 1 means that each segment behave differently. The full specification of each of those segments must therefore be given.
  - Segment-cabin section
  - Fare family section
- Segment-cabin section
  - Cabin code (e.g., F, J, W or Y)
  - List of (one-letter-code) booking classes for the cabin (e.g, respectively FA, JC DI, WT or YBHKMLSQ)
- Fare family section
  - Fare family code (e.g., 1)
  - List of (one-letter-code) booking classes for the fare family (e.g, respectively FA, JC DI, WT or YBHKMLSQ)

Some fare input examples (including the example above named schedule03.csv) are given in the StdAir project.

### 12.12.5 The schedule BOM Tree

The schedule-related Business Object Model (BOM) tree is a structure allowing to store all the `AIRTSP::FlightPeriodStruct` objects of the simulation. That is why parsing an input file, containing the specification for all the flight-periods, is more convenient (

#### See also:

the previous section [Input file of AirTSP Project](#)).

As it may be time consuming, and it for sure requires some know-how, to first build such a schedule input file, a small sample BOM tree is provided by default when needed.

**12.12.5.1 Build of the schedule BOM tree** 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 (during the instantiation of the `AIRTSP::AIRTSP_Service` object).

The corresponding type (class) `stdair::BomRoot` is defined in the `StdAir` library.

Then, the BOM root can be either constructed thanks to the `AIRTSP::AIRTSP_Service::buildSampleBom()` method:

```
void buildSampleBom();
```

or can be constructed using the schedule input file described above thanks to the `AIRTSP::AIRTSP_Service::parseAndLoad` (`const stdair::Filename_T&`) method:

```
void parseAndLoad (const stdair::ScheduleFilePath&);
```

### 12.12.5.2 Display of the schedule BOM tree

**Note:**

That feature (of BOM tree display) has not been implemented yet. Do not hesitate to [open a ticket](#) if you would like to have it implemented more quickly.

The schedule BOM tree can be displayed as done in the `batches::airtsp.cpp` program:

When the default BOM tree is used (`-b/-builtin` option of the main program `airtsp.cpp`), the schedule BOM tree display (for now, corresponding to `schedule01.csv` parsed by `AIRINV::parseInventory`) should look like:

```
=====
BomRoot:  -- ROOT  --
=====
+++++
Inventory: SQ
+++++
*****
FlightDate: SQ11, 2010-Jan-15
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-15, SIN-BKK, 2010-Jan-15, 08:20:00, 2010-Jan-15, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-15, SIN-BKK 2010-Jan-15, Y, 300, 300, 0, 0, 0, 0, 0, 0, 2, 298, 9, 0, 0, 0, 0,
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
```

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ11 2010-Jan-15, SIN-BKK 2010-Jan-15, Y, 1, 0, 0, 0, 2, 298, 0,  
SQ11 2010-Jan-15, SIN-BKK 2010-Jan-15, Y, 2, 0, 0, 0, 2, 298, 0,

\*\*\*\*\*

\*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ11 2010-Jan-15, SIN-BKK 2010-Jan-15, Y, 1, Y, 300 (0), 0, 0, 0, 2, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ11 2010-Jan-15, SIN-BKK 2010-Jan-15, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

FlightDate: SQ11, 2010-Jan-16

\*\*\*\*\*

\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ11 2010-Jan-16, SIN-BKK, 2010-Jan-16, 08:20:00, 2010-Jan-16, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*

\*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ11 2010-Jan-16, SIN-BKK 2010-Jan-16, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 1.83244e-319, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*  
\*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ11 2010-Jan-16, SIN-BKK 2010-Jan-16, Y, 1, 0, 0, 0, 0, 300, 0,  
SQ11 2010-Jan-16, SIN-BKK 2010-Jan-16, Y, 2, 0, 0, 0, 0, 300, 0,

\*\*\*\*\*

\*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ11 2010-Jan-16, SIN-BKK 2010-Jan-16, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ11 2010-Jan-16, SIN-BKK 2010-Jan-16, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

FlightDate: SQ11, 2010-Jan-17

\*\*\*\*\*

\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ11 2010-Jan-17, SIN-BKK, 2010-Jan-17, 08:20:00, 2010-Jan-17, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*

\*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ11 2010-Jan-17, SIN-BKK 2010-Jan-17, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 1.58896e-319, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*

\*\*\*\*\*  
SegmentCabins:  
-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ11 2010-Jan-17, SIN-BKK 2010-Jan-17, Y, 1, 0, 0, 0, 0, 300, 0,  
SQ11 2010-Jan-17, SIN-BKK 2010-Jan-17, Y, 2, 0, 0, 0, 0, 300, 0,

\*\*\*\*\*  
\*\*\*\*\*  
Subclasses:  
-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ11 2010-Jan-17, SIN-BKK 2010-Jan-17, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ11 2010-Jan-17, SIN-BKK 2010-Jan-17, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*  
FlightDate: SQ11, 2010-Jan-18  
\*\*\*\*\*

\*\*\*\*\*  
\*\*\*\*\*  
Leg-Dates:  
-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ11 2010-Jan-18, SIN-BKK, 2010-Jan-18, 08:20:00, 2010-Jan-18, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*  
\*\*\*\*\*  
LegCabins:  
-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ11 2010-Jan-18, SIN-BKK 2010-Jan-18, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*  
Buckets:  
-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*  
\*\*\*\*\*  
SegmentCabins:  
-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ11 2010-Jan-18, SIN-BKK 2010-Jan-18, Y, 1, 0, 0, 0, 0, 300, 0,  
SQ11 2010-Jan-18, SIN-BKK 2010-Jan-18, Y, 2, 0, 0, 0, 0, 300, 0,

\*\*\*\*\*  
\*\*\*\*\*  
Subclasses:  
-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ11 2010-Jan-18, SIN-BKK 2010-Jan-18, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ11 2010-Jan-18, SIN-BKK 2010-Jan-18, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*  
FlightDate: SQ11, 2010-Jan-19  
\*\*\*\*\*

\*\*\*\*\*  
\*\*\*\*\*  
Leg-Dates:  
-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ11 2010-Jan-19, SIN-BKK, 2010-Jan-19, 08:20:00, 2010-Jan-19, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*  
\*\*\*\*\*  
LegCabins:  
-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ11 2010-Jan-19, SIN-BKK 2010-Jan-19, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*  
Buckets:  
-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

```
*****
*****
SegmentCabins:
-----
```

```
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-19, SIN-BKK 2010-Jan-19, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-19, SIN-BKK 2010-Jan-19, Y, 2, 0, 0, 0, 0, 300, 0,
```

```
*****
*****
Subclasses:
-----
```

```
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-19, SIN-BKK 2010-Jan-19, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0,
SQ11 2010-Jan-19, SIN-BKK 2010-Jan-19, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
```

```
*****
*****
FlightDate: SQ11, 2010-Jan-20
*****
*****
```

```
Leg-Dates:
-----
```

```
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-20, SIN-BKK, 2010-Jan-20, 08:20:00, 2010-Jan-20, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
```

```
*****
*****
LegCabins:
-----
```

```
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-20, SIN-BKK 2010-Jan-20, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
```

```
*****
*****
Buckets:
-----
```

```
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
```

```
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-20, SIN-BKK 2010-Jan-20, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-20, SIN-BKK 2010-Jan-20, Y, 2, 0, 0, 0, 0, 300, 0,
```

```
*****
*****
Subclasses:
-----
```

```
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-20, SIN-BKK 2010-Jan-20, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Jan-20, SIN-BKK 2010-Jan-20, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
```

```
*****
*****
FlightDate: SQ11, 2010-Jan-21
*****
*****
```

```
Leg-Dates:
-----
```

```
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-21, SIN-BKK, 2010-Jan-21, 08:20:00, 2010-Jan-21, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
```

```
*****
*****
LegCabins:
-----
```

```
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-21, SIN-BKK 2010-Jan-21, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
```

```
*****
*****
Buckets:
-----
```



```
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-21, SIN-BKK 2010-Jan-21, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-21, SIN-BKK 2010-Jan-21, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-21, SIN-BKK 2010-Jan-21, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Jan-21, SIN-BKK 2010-Jan-21, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Jan-22
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-22, SIN-BKK, 2010-Jan-22, 08:20:00, 2010-Jan-22, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-22, SIN-BKK 2010-Jan-22, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-22, SIN-BKK 2010-Jan-22, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-22, SIN-BKK 2010-Jan-22, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-22, SIN-BKK 2010-Jan-22, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Jan-22, SIN-BKK 2010-Jan-22, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Jan-23
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-23, SIN-BKK, 2010-Jan-23, 08:20:00, 2010-Jan-23, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-23, SIN-BKK 2010-Jan-23, Y, 300, 300, 0, 0, 0, 0, 0, 0, 6.64029e-319, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
```

```
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-23, SIN-BKK 2010-Jan-23, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-23, SIN-BKK 2010-Jan-23, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-23, SIN-BKK 2010-Jan-23, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Jan-23, SIN-BKK 2010-Jan-23, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Jan-24
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-24, SIN-BKK, 2010-Jan-24, 08:20:00, 2010-Jan-24, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-24, SIN-BKK 2010-Jan-24, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-24, SIN-BKK 2010-Jan-24, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-24, SIN-BKK 2010-Jan-24, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-24, SIN-BKK 2010-Jan-24, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Jan-24, SIN-BKK 2010-Jan-24, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Jan-25
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-25, SIN-BKK, 2010-Jan-25, 08:20:00, 2010-Jan-25, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-25, SIN-BKK 2010-Jan-25, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
```

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
 \*\*\*\*\*  
 \*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
 SQ11 2010-Jan-25, SIN-BKK 2010-Jan-25, Y, 1, 0, 0, 0, 0, 300, 0,  
 SQ11 2010-Jan-25, SIN-BKK 2010-Jan-25, Y, 2, 0, 0, 0, 0, 300, 0,  
 \*\*\*\*\*  
 \*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
 SQ11 2010-Jan-25, SIN-BKK 2010-Jan-25, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
 SQ11 2010-Jan-25, SIN-BKK 2010-Jan-25, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
 \*\*\*\*\*  
 \*\*\*\*\*

FlightDate: SQ11, 2010-Jan-26

\*\*\*\*\*  
 \*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
 SQ11 2010-Jan-26, SIN-BKK, 2010-Jan-26, 08:20:00, 2010-Jan-26, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,  
 \*\*\*\*\*  
 \*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
 SQ11 2010-Jan-26, SIN-BKK 2010-Jan-26, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,  
 \*\*\*\*\*  
 \*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
 \*\*\*\*\*  
 \*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
 SQ11 2010-Jan-26, SIN-BKK 2010-Jan-26, Y, 1, 0, 0, 0, 0, 300, 0,  
 SQ11 2010-Jan-26, SIN-BKK 2010-Jan-26, Y, 2, 0, 0, 0, 0, 300, 0,  
 \*\*\*\*\*  
 \*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
 SQ11 2010-Jan-26, SIN-BKK 2010-Jan-26, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
 SQ11 2010-Jan-26, SIN-BKK 2010-Jan-26, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
 \*\*\*\*\*  
 \*\*\*\*\*

FlightDate: SQ11, 2010-Jan-27

\*\*\*\*\*  
 \*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
 SQ11 2010-Jan-27, SIN-BKK, 2010-Jan-27, 08:20:00, 2010-Jan-27, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,  
 \*\*\*\*\*  
 \*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
 SQ11 2010-Jan-27, SIN-BKK 2010-Jan-27, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,  
 \*\*\*\*\*

\*\*\*\*\*  
Buckets:  
-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*  
\*\*\*\*\*  
SegmentCabins:  
-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ11 2010-Jan-27, SIN-BKK 2010-Jan-27, Y, 1, 0, 0, 0, 0, 300, 0,  
SQ11 2010-Jan-27, SIN-BKK 2010-Jan-27, Y, 2, 0, 0, 0, 0, 300, 0,  
\*\*\*\*\*  
\*\*\*\*\*  
Subclasses:  
-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ11 2010-Jan-27, SIN-BKK 2010-Jan-27, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ11 2010-Jan-27, SIN-BKK 2010-Jan-27, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
\*\*\*\*\*  
\*\*\*\*\*  
FlightDate: SQ11, 2010-Jan-28  
\*\*\*\*\*  
\*\*\*\*\*

Leg-Dates:  
-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ11 2010-Jan-28, SIN-BKK, 2010-Jan-28, 08:20:00, 2010-Jan-28, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,  
\*\*\*\*\*  
\*\*\*\*\*

LegCabins:  
-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ11 2010-Jan-28, SIN-BKK 2010-Jan-28, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,  
\*\*\*\*\*  
\*\*\*\*\*

Buckets:  
-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*  
\*\*\*\*\*  
SegmentCabins:  
-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ11 2010-Jan-28, SIN-BKK 2010-Jan-28, Y, 1, 0, 0, 0, 0, 300, 0,  
SQ11 2010-Jan-28, SIN-BKK 2010-Jan-28, Y, 2, 0, 0, 0, 0, 300, 0,  
\*\*\*\*\*  
\*\*\*\*\*  
Subclasses:  
-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ11 2010-Jan-28, SIN-BKK 2010-Jan-28, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ11 2010-Jan-28, SIN-BKK 2010-Jan-28, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
\*\*\*\*\*  
\*\*\*\*\*  
FlightDate: SQ11, 2010-Jan-29  
\*\*\*\*\*  
\*\*\*\*\*

Leg-Dates:  
-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ11 2010-Jan-29, SIN-BKK, 2010-Jan-29, 08:20:00, 2010-Jan-29, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,  
\*\*\*\*\*  
\*\*\*\*\*

LegCabins:  
-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ11 2010-Jan-29, SIN-BKK 2010-Jan-29, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

```
*****
*****
Buckets:
-----
```

```
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
```

```
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-29, SIN-BKK 2010-Jan-29, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-29, SIN-BKK 2010-Jan-29, Y, 2, 0, 0, 0, 0, 300, 0,
```

```
*****
*****
Subclasses:
-----
```

```
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-29, SIN-BKK 2010-Jan-29, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Jan-29, SIN-BKK 2010-Jan-29, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
```

```
*****
*****
FlightDate: SQ11, 2010-Jan-30
*****
*****
```

```
Leg-Dates:
-----
```

```
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-30, SIN-BKK, 2010-Jan-30, 08:20:00, 2010-Jan-30, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
```

```
LegCabins:
-----
```

```
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-30, SIN-BKK 2010-Jan-30, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
```

```
*****
*****
Buckets:
-----
```

```
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
```

```
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-30, SIN-BKK 2010-Jan-30, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-30, SIN-BKK 2010-Jan-30, Y, 2, 0, 0, 0, 0, 300, 0,
```

```
*****
*****
Subclasses:
-----
```

```
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-30, SIN-BKK 2010-Jan-30, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Jan-30, SIN-BKK 2010-Jan-30, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
```

```
*****
*****
FlightDate: SQ11, 2010-Jan-31
*****
*****
```

```
Leg-Dates:
-----
```

```
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-31, SIN-BKK, 2010-Jan-31, 08:20:00, 2010-Jan-31, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
```

```
LegCabins:
-----
```

```
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
```

```
SQL1 2010-Jan-31, SIN-BKK 2010-Jan-31, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL1 2010-Jan-31, SIN-BKK 2010-Jan-31, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 2010-Jan-31, SIN-BKK 2010-Jan-31, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQL1 2010-Jan-31, SIN-BKK 2010-Jan-31, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQL1 2010-Jan-31, SIN-BKK 2010-Jan-31, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL1, 2010-Feb-01
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 2010-Feb-01, SIN-BKK, 2010-Feb-01, 08:20:00, 2010-Feb-01, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL1 2010-Feb-01, SIN-BKK 2010-Feb-01, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL1 2010-Feb-01, SIN-BKK 2010-Feb-01, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 2010-Feb-01, SIN-BKK 2010-Feb-01, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQL1 2010-Feb-01, SIN-BKK 2010-Feb-01, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQL1 2010-Feb-01, SIN-BKK 2010-Feb-01, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL1, 2010-Feb-02
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 2010-Feb-02, SIN-BKK, 2010-Feb-02, 08:20:00, 2010-Feb-02, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
```

```
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-02, SIN-BKK 2010-Feb-02, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-02, SIN-BKK 2010-Feb-02, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-02, SIN-BKK 2010-Feb-02, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-02, SIN-BKK 2010-Feb-02, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-02, SIN-BKK 2010-Feb-02, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-03
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-03, SIN-BKK, 2010-Feb-03, 08:20:00, 2010-Feb-03, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-03, SIN-BKK 2010-Feb-03, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-03, SIN-BKK 2010-Feb-03, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-03, SIN-BKK 2010-Feb-03, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-03, SIN-BKK 2010-Feb-03, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-03, SIN-BKK 2010-Feb-03, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-04
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-04, SIN-BKK, 2010-Feb-04, 08:20:00, 2010-Feb-04, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
```

```
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-04, SIN-BKK 2010-Feb-04, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-04, SIN-BKK 2010-Feb-04, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-04, SIN-BKK 2010-Feb-04, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-04, SIN-BKK 2010-Feb-04, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-04, SIN-BKK 2010-Feb-04, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-05
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-05, SIN-BKK, 2010-Feb-05, 08:20:00, 2010-Feb-05, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-05, SIN-BKK 2010-Feb-05, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-05, SIN-BKK 2010-Feb-05, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-05, SIN-BKK 2010-Feb-05, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-05, SIN-BKK 2010-Feb-05, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-05, SIN-BKK 2010-Feb-05, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-06
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-06, SIN-BKK, 2010-Feb-06, 08:20:00, 2010-Feb-06, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
```



LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ11 2010-Feb-06, SIN-BKK 2010-Feb-06, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,  
\*\*\*\*\*  
\*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*  
\*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ11 2010-Feb-06, SIN-BKK 2010-Feb-06, Y, 1, 0, 0, 0, 0, 300, 0,  
SQ11 2010-Feb-06, SIN-BKK 2010-Feb-06, Y, 2, 0, 0, 0, 0, 300, 0,  
\*\*\*\*\*  
\*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ11 2010-Feb-06, SIN-BKK 2010-Feb-06, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ11 2010-Feb-06, SIN-BKK 2010-Feb-06, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
\*\*\*\*\*  
\*\*\*\*\*

FlightDate: SQ11, 2010-Feb-07

\*\*\*\*\*  
\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ11 2010-Feb-07, SIN-BKK, 2010-Feb-07, 08:20:00, 2010-Feb-07, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,  
\*\*\*\*\*  
\*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ11 2010-Feb-07, SIN-BKK 2010-Feb-07, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,  
\*\*\*\*\*  
\*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*  
\*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ11 2010-Feb-07, SIN-BKK 2010-Feb-07, Y, 1, 0, 0, 0, 0, 300, 0,  
SQ11 2010-Feb-07, SIN-BKK 2010-Feb-07, Y, 2, 0, 0, 0, 0, 300, 0,  
\*\*\*\*\*  
\*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ11 2010-Feb-07, SIN-BKK 2010-Feb-07, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ11 2010-Feb-07, SIN-BKK 2010-Feb-07, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
\*\*\*\*\*  
\*\*\*\*\*

FlightDate: SQ11, 2010-Feb-08

\*\*\*\*\*  
\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ11 2010-Feb-08, SIN-BKK, 2010-Feb-08, 08:20:00, 2010-Feb-08, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,  
\*\*\*\*\*  
\*\*\*\*\*

\*\*\*\*\*  
LegCabins:  
-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ11 2010-Feb-08, SIN-BKK 2010-Feb-08, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*  
Buckets:  
-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*  
\*\*\*\*\*

SegmentCabins:  
-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ11 2010-Feb-08, SIN-BKK 2010-Feb-08, Y, 1, 0, 0, 0, 0, 300, 0,  
SQ11 2010-Feb-08, SIN-BKK 2010-Feb-08, Y, 2, 0, 0, 0, 0, 300, 0,

\*\*\*\*\*  
\*\*\*\*\*  
Subclasses:  
-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ11 2010-Feb-08, SIN-BKK 2010-Feb-08, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ11 2010-Feb-08, SIN-BKK 2010-Feb-08, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*  
FlightDate: SQ11, 2010-Feb-09  
\*\*\*\*\*  
\*\*\*\*\*

Leg-Dates:  
-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ11 2010-Feb-09, SIN-BKK, 2010-Feb-09, 08:20:00, 2010-Feb-09, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*  
\*\*\*\*\*  
LegCabins:  
-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ11 2010-Feb-09, SIN-BKK 2010-Feb-09, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*  
Buckets:  
-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*  
\*\*\*\*\*

SegmentCabins:  
-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ11 2010-Feb-09, SIN-BKK 2010-Feb-09, Y, 1, 0, 0, 0, 0, 300, 0,  
SQ11 2010-Feb-09, SIN-BKK 2010-Feb-09, Y, 2, 0, 0, 0, 0, 300, 0,

\*\*\*\*\*  
\*\*\*\*\*  
Subclasses:  
-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ11 2010-Feb-09, SIN-BKK 2010-Feb-09, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ11 2010-Feb-09, SIN-BKK 2010-Feb-09, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*  
FlightDate: SQ11, 2010-Feb-10  
\*\*\*\*\*  
\*\*\*\*\*

Leg-Dates:  
-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ11 2010-Feb-10, SIN-BKK, 2010-Feb-10, 08:20:00, 2010-Feb-10, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

```
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-10, SIN-BKK 2010-Feb-10, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-10, SIN-BKK 2010-Feb-10, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-10, SIN-BKK 2010-Feb-10, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-10, SIN-BKK 2010-Feb-10, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-10, SIN-BKK 2010-Feb-10, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-11
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-11, SIN-BKK, 2010-Feb-11, 08:20:00, 2010-Feb-11, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-11, SIN-BKK 2010-Feb-11, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-11, SIN-BKK 2010-Feb-11, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-11, SIN-BKK 2010-Feb-11, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-11, SIN-BKK 2010-Feb-11, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-11, SIN-BKK 2010-Feb-11, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-12
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
```

```
SQL1 2010-Feb-12, SIN-BKK, 2010-Feb-12, 08:20:00, 2010-Feb-12, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL1 2010-Feb-12, SIN-BKK 2010-Feb-12, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL1 2010-Feb-12, SIN-BKK 2010-Feb-12, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 2010-Feb-12, SIN-BKK 2010-Feb-12, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQL1 2010-Feb-12, SIN-BKK 2010-Feb-12, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQL1 2010-Feb-12, SIN-BKK 2010-Feb-12, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL1, 2010-Feb-13
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 2010-Feb-13, SIN-BKK, 2010-Feb-13, 08:20:00, 2010-Feb-13, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL1 2010-Feb-13, SIN-BKK 2010-Feb-13, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL1 2010-Feb-13, SIN-BKK 2010-Feb-13, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 2010-Feb-13, SIN-BKK 2010-Feb-13, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQL1 2010-Feb-13, SIN-BKK 2010-Feb-13, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQL1 2010-Feb-13, SIN-BKK 2010-Feb-13, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL1, 2010-Feb-14
*****
*****
Leg-Dates:
-----
```

```
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-14, SIN-BKK, 2010-Feb-14, 08:20:00, 2010-Feb-14, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-14, SIN-BKK 2010-Feb-14, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-14, SIN-BKK 2010-Feb-14, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-14, SIN-BKK 2010-Feb-14, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-14, SIN-BKK 2010-Feb-14, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-14, SIN-BKK 2010-Feb-14, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-15
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-15, SIN-BKK, 2010-Feb-15, 08:20:00, 2010-Feb-15, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-15, SIN-BKK 2010-Feb-15, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-15, SIN-BKK 2010-Feb-15, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-15, SIN-BKK 2010-Feb-15, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-15, SIN-BKK 2010-Feb-15, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-15, SIN-BKK 2010-Feb-15, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-16
*****
*****
Leg-Dates:
```

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
 SQ11 2010-Feb-16, SIN-BKK, 2010-Feb-16, 08:20:00, 2010-Feb-16, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,  
 \*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
 SQ11 2010-Feb-16, SIN-BKK 2010-Feb-16, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,  
 \*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
 \*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
 SQ11 2010-Feb-16, SIN-BKK 2010-Feb-16, Y, 1, 0, 0, 0, 0, 300, 0,  
 SQ11 2010-Feb-16, SIN-BKK 2010-Feb-16, Y, 2, 0, 0, 0, 0, 300, 0,  
 \*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
 SQ11 2010-Feb-16, SIN-BKK 2010-Feb-16, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
 SQ11 2010-Feb-16, SIN-BKK 2010-Feb-16, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
 \*\*\*\*\*

FlightDate: SQ11, 2010-Feb-17

\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
 SQ11 2010-Feb-17, SIN-BKK, 2010-Feb-17, 08:20:00, 2010-Feb-17, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,  
 \*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
 SQ11 2010-Feb-17, SIN-BKK 2010-Feb-17, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,  
 \*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
 \*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
 SQ11 2010-Feb-17, SIN-BKK 2010-Feb-17, Y, 1, 0, 0, 0, 0, 300, 0,  
 SQ11 2010-Feb-17, SIN-BKK 2010-Feb-17, Y, 2, 0, 0, 0, 0, 300, 0,  
 \*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
 SQ11 2010-Feb-17, SIN-BKK 2010-Feb-17, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
 SQ11 2010-Feb-17, SIN-BKK 2010-Feb-17, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
 \*\*\*\*\*

FlightDate: SQ11, 2010-Feb-18

\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ11 2010-Feb-18, SIN-BKK, 2010-Feb-18, 08:20:00, 2010-Feb-18, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,  
\*\*\*\*\*  
\*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ11 2010-Feb-18, SIN-BKK 2010-Feb-18, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,  
\*\*\*\*\*  
\*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*  
\*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ11 2010-Feb-18, SIN-BKK 2010-Feb-18, Y, 1, 0, 0, 0, 0, 300, 0,  
SQ11 2010-Feb-18, SIN-BKK 2010-Feb-18, Y, 2, 0, 0, 0, 0, 300, 0,  
\*\*\*\*\*  
\*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ11 2010-Feb-18, SIN-BKK 2010-Feb-18, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ11 2010-Feb-18, SIN-BKK 2010-Feb-18, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
\*\*\*\*\*  
\*\*\*\*\*

FlightDate: SQ11, 2010-Feb-19

\*\*\*\*\*  
\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ11 2010-Feb-19, SIN-BKK, 2010-Feb-19, 08:20:00, 2010-Feb-19, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,  
\*\*\*\*\*  
\*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ11 2010-Feb-19, SIN-BKK 2010-Feb-19, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,  
\*\*\*\*\*  
\*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*  
\*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ11 2010-Feb-19, SIN-BKK 2010-Feb-19, Y, 1, 0, 0, 0, 0, 300, 0,  
SQ11 2010-Feb-19, SIN-BKK 2010-Feb-19, Y, 2, 0, 0, 0, 0, 300, 0,  
\*\*\*\*\*  
\*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ11 2010-Feb-19, SIN-BKK 2010-Feb-19, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ11 2010-Feb-19, SIN-BKK 2010-Feb-19, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
\*\*\*\*\*  
\*\*\*\*\*

FlightDate: SQ11, 2010-Feb-20

\*\*\*\*\*

\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ11 2010-Feb-20, SIN-BKK, 2010-Feb-20, 08:20:00, 2010-Feb-20, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*

\*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ11 2010-Feb-20, SIN-BKK 2010-Feb-20, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

\*\*\*\*\*

\*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,

SQ11 2010-Feb-20, SIN-BKK 2010-Feb-20, Y, 1, 0, 0, 0, 0, 300, 0,

SQ11 2010-Feb-20, SIN-BKK 2010-Feb-20, Y, 2, 0, 0, 0, 0, 300, 0,

\*\*\*\*\*

\*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,

SQ11 2010-Feb-20, SIN-BKK 2010-Feb-20, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

SQ11 2010-Feb-20, SIN-BKK 2010-Feb-20, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

FlightDate: SQ11, 2010-Feb-21

\*\*\*\*\*

\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity

SQ11 2010-Feb-21, SIN-BKK, 2010-Feb-21, 08:20:00, 2010-Feb-21, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*

\*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,

SQ11 2010-Feb-21, SIN-BKK 2010-Feb-21, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

\*\*\*\*\*

\*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,

SQ11 2010-Feb-21, SIN-BKK 2010-Feb-21, Y, 1, 0, 0, 0, 0, 300, 0,

SQ11 2010-Feb-21, SIN-BKK 2010-Feb-21, Y, 2, 0, 0, 0, 0, 300, 0,

\*\*\*\*\*

\*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,

SQ11 2010-Feb-21, SIN-BKK 2010-Feb-21, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

SQ11 2010-Feb-21, SIN-BKK 2010-Feb-21, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

FlightDate: SQ11, 2010-Feb-22



```
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-22, SIN-BKK, 2010-Feb-22, 08:20:00, 2010-Feb-22, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-22, SIN-BKK 2010-Feb-22, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-22, SIN-BKK 2010-Feb-22, Y, 1, 0, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-22, SIN-BKK 2010-Feb-22, Y, 2, 0, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-22, SIN-BKK 2010-Feb-22, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-22, SIN-BKK 2010-Feb-22, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-23
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-23, SIN-BKK, 2010-Feb-23, 08:20:00, 2010-Feb-23, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-23, SIN-BKK 2010-Feb-23, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-23, SIN-BKK 2010-Feb-23, Y, 1, 0, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-23, SIN-BKK 2010-Feb-23, Y, 2, 0, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-23, SIN-BKK 2010-Feb-23, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-23, SIN-BKK 2010-Feb-23, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
```

FlightDate: SQ11, 2010-Feb-24

\*\*\*\*\*  
\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity

SQ11 2010-Feb-24, SIN-BKK, 2010-Feb-24, 08:20:00, 2010-Feb-24, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*  
\*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,

SQ11 2010-Feb-24, SIN-BKK 2010-Feb-24, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

\*\*\*\*\*  
\*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,

SQ11 2010-Feb-24, SIN-BKK 2010-Feb-24, Y, 1, 0, 0, 0, 0, 300, 0,

SQ11 2010-Feb-24, SIN-BKK 2010-Feb-24, Y, 2, 0, 0, 0, 0, 300, 0,

\*\*\*\*\*  
\*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,

SQ11 2010-Feb-24, SIN-BKK 2010-Feb-24, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

SQ11 2010-Feb-24, SIN-BKK 2010-Feb-24, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*

FlightDate: SQ11, 2010-Feb-25

\*\*\*\*\*  
\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity

SQ11 2010-Feb-25, SIN-BKK, 2010-Feb-25, 08:20:00, 2010-Feb-25, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*  
\*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,

SQ11 2010-Feb-25, SIN-BKK 2010-Feb-25, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

\*\*\*\*\*  
\*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,

SQ11 2010-Feb-25, SIN-BKK 2010-Feb-25, Y, 1, 0, 0, 0, 0, 300, 0,

SQ11 2010-Feb-25, SIN-BKK 2010-Feb-25, Y, 2, 0, 0, 0, 0, 300, 0,

\*\*\*\*\*  
\*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,

SQ11 2010-Feb-25, SIN-BKK 2010-Feb-25, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

SQ11 2010-Feb-25, SIN-BKK 2010-Feb-25, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*  
FlightDate: SQ11, 2010-Feb-26  
\*\*\*\*\*

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ11 2010-Feb-26, SIN-BKK, 2010-Feb-26, 08:20:00, 2010-Feb-26, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ11 2010-Feb-26, SIN-BKK 2010-Feb-26, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ11 2010-Feb-26, SIN-BKK 2010-Feb-26, Y, 1, 0, 0, 0, 0, 300, 0,  
SQ11 2010-Feb-26, SIN-BKK 2010-Feb-26, Y, 2, 0, 0, 0, 0, 300, 0,

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ11 2010-Feb-26, SIN-BKK 2010-Feb-26, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ11 2010-Feb-26, SIN-BKK 2010-Feb-26, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

FlightDate: SQ11, 2010-Feb-27

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ11 2010-Feb-27, SIN-BKK, 2010-Feb-27, 08:20:00, 2010-Feb-27, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ11 2010-Feb-27, SIN-BKK 2010-Feb-27, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ11 2010-Feb-27, SIN-BKK 2010-Feb-27, Y, 1, 0, 0, 0, 0, 300, 0,  
SQ11 2010-Feb-27, SIN-BKK 2010-Feb-27, Y, 2, 0, 0, 0, 0, 300, 0,

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ11 2010-Feb-27, SIN-BKK 2010-Feb-27, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ11 2010-Feb-27, SIN-BKK 2010-Feb-27, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

```
*****
*****
FlightDate: SQ11, 2010-Feb-28
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-28, SIN-BKK, 2010-Feb-28, 08:20:00, 2010-Feb-28, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-28, SIN-BKK 2010-Feb-28, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-28, SIN-BKK 2010-Feb-28, Y, 1, 0, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-28, SIN-BKK 2010-Feb-28, Y, 2, 0, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-28, SIN-BKK 2010-Feb-28, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-28, SIN-BKK 2010-Feb-28, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Jan-15
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Jan-15, SIN-HND, 2010-Jan-15, 09:20:00, 2010-Jan-15, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Jan-15, SIN-HND 2010-Jan-15, Y, 200, 200, 2.082e+121, 5.53287e-48, 5.20268e-90, 0, 1.31346e-47,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Jan-15, SIN-HND 2010-Jan-15, Y, 1, 0, 0, 0, 0, 0, 200, 0,
SQ12 2010-Jan-15, SIN-HND 2010-Jan-15, Y, 2, 0, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Jan-15, SIN-HND 2010-Jan-15, Y, 1, Y13856, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
```

```
SQL2 2010-Jan-15, SIN-HND 2010-Jan-15, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL2, 2010-Jan-16
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 2010-Jan-16, SIN-HND, 2010-Jan-16, 09:20:00, 2010-Jan-16, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL2 2010-Jan-16, SIN-HND 2010-Jan-16, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 2.63638e-319, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL2 2010-Jan-16, SIN-HND 2010-Jan-16, Y, 1, 0, 0, 0, 0, 200, 0,
SQL2 2010-Jan-16, SIN-HND 2010-Jan-16, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQL2 2010-Jan-16, SIN-HND 2010-Jan-16, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQL2 2010-Jan-16, SIN-HND 2010-Jan-16, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL2, 2010-Jan-17
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 2010-Jan-17, SIN-HND, 2010-Jan-17, 09:20:00, 2010-Jan-17, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL2 2010-Jan-17, SIN-HND 2010-Jan-17, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 2.39291e-319, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL2 2010-Jan-17, SIN-HND 2010-Jan-17, Y, 1, 0, 0, 0, 0, 200, 0,
SQL2 2010-Jan-17, SIN-HND 2010-Jan-17, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
```

```
SQL2 2010-Jan-17, SIN-HND 2010-Jan-17, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQL2 2010-Jan-17, SIN-HND 2010-Jan-17, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL2, 2010-Jan-18
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 2010-Jan-18, SIN-HND, 2010-Jan-18, 09:20:00, 2010-Jan-18, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL2 2010-Jan-18, SIN-HND 2010-Jan-18, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 2.14469e-319, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL2 2010-Jan-18, SIN-HND 2010-Jan-18, Y, 1, 0, 0, 0, 0, 200, 0,
SQL2 2010-Jan-18, SIN-HND 2010-Jan-18, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQL2 2010-Jan-18, SIN-HND 2010-Jan-18, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQL2 2010-Jan-18, SIN-HND 2010-Jan-18, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL2, 2010-Jan-19
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 2010-Jan-19, SIN-HND, 2010-Jan-19, 09:20:00, 2010-Jan-19, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL2 2010-Jan-19, SIN-HND 2010-Jan-19, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL2 2010-Jan-19, SIN-HND 2010-Jan-19, Y, 1, 0, 0, 0, 0, 200, 0,
SQL2 2010-Jan-19, SIN-HND 2010-Jan-19, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
```

```
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Jan-19, SIN-HND 2010-Jan-19, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Jan-19, SIN-HND 2010-Jan-19, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Jan-20
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Jan-20, SIN-HND, 2010-Jan-20, 09:20:00, 2010-Jan-20, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Jan-20, SIN-HND 2010-Jan-20, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Jan-20, SIN-HND 2010-Jan-20, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Jan-20, SIN-HND 2010-Jan-20, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Jan-20, SIN-HND 2010-Jan-20, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Jan-20, SIN-HND 2010-Jan-20, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Jan-21
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Jan-21, SIN-HND, 2010-Jan-21, 09:20:00, 2010-Jan-21, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Jan-21, SIN-HND 2010-Jan-21, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Jan-21, SIN-HND 2010-Jan-21, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Jan-21, SIN-HND 2010-Jan-21, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
```

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ12 2010-Jan-21, SIN-HND 2010-Jan-21, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ12 2010-Jan-21, SIN-HND 2010-Jan-21, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

FlightDate: SQ12, 2010-Jan-22

\*\*\*\*\*

\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ12 2010-Jan-22, SIN-HND, 2010-Jan-22, 09:20:00, 2010-Jan-22, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*

\*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ12 2010-Jan-22, SIN-HND 2010-Jan-22, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

\*\*\*\*\*

\*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,

SQ12 2010-Jan-22, SIN-HND 2010-Jan-22, Y, 1, 0, 0, 0, 0, 200, 0,

SQ12 2010-Jan-22, SIN-HND 2010-Jan-22, Y, 2, 0, 0, 0, 0, 200, 0,

\*\*\*\*\*

\*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ12 2010-Jan-22, SIN-HND 2010-Jan-22, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ12 2010-Jan-22, SIN-HND 2010-Jan-22, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

FlightDate: SQ12, 2010-Jan-23

\*\*\*\*\*

\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ12 2010-Jan-23, SIN-HND, 2010-Jan-23, 09:20:00, 2010-Jan-23, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*

\*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ12 2010-Jan-23, SIN-HND 2010-Jan-23, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

\*\*\*\*\*

\*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,

SQ12 2010-Jan-23, SIN-HND 2010-Jan-23, Y, 1, 0, 0, 0, 0, 200, 0,

SQ12 2010-Jan-23, SIN-HND 2010-Jan-23, Y, 2, 0, 0, 0, 0, 200, 0,

\*\*\*\*\*

\*\*\*\*\*



Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ12 2010-Jan-23, SIN-HND 2010-Jan-23, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ12 2010-Jan-23, SIN-HND 2010-Jan-23, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

FlightDate: SQ12, 2010-Jan-24

\*\*\*\*\*

\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ12 2010-Jan-24, SIN-HND, 2010-Jan-24, 09:20:00, 2010-Jan-24, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*

\*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ12 2010-Jan-24, SIN-HND 2010-Jan-24, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

\*\*\*\*\*

\*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,

SQ12 2010-Jan-24, SIN-HND 2010-Jan-24, Y, 1, 0, 0, 0, 0, 200, 0,

SQ12 2010-Jan-24, SIN-HND 2010-Jan-24, Y, 2, 0, 0, 0, 0, 200, 0,

\*\*\*\*\*

\*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ12 2010-Jan-24, SIN-HND 2010-Jan-24, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ12 2010-Jan-24, SIN-HND 2010-Jan-24, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

FlightDate: SQ12, 2010-Jan-25

\*\*\*\*\*

\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity

SQ12 2010-Jan-25, SIN-HND, 2010-Jan-25, 09:20:00, 2010-Jan-25, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*

\*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ12 2010-Jan-25, SIN-HND 2010-Jan-25, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

\*\*\*\*\*

\*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,

SQ12 2010-Jan-25, SIN-HND 2010-Jan-25, Y, 1, 0, 0, 0, 0, 200, 0,

SQ12 2010-Jan-25, SIN-HND 2010-Jan-25, Y, 2, 0, 0, 0, 0, 200, 0,

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*  
Subclasses:  
-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ12 2010-Jan-25, SIN-HND 2010-Jan-25, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ12 2010-Jan-25, SIN-HND 2010-Jan-25, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*  
FlightDate: SQ12, 2010-Jan-26  
\*\*\*\*\*

\*\*\*\*\*  
Leg-Dates:  
-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ12 2010-Jan-26, SIN-HND, 2010-Jan-26, 09:20:00, 2010-Jan-26, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*  
\*\*\*\*\*  
LegCabins:  
-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ12 2010-Jan-26, SIN-HND 2010-Jan-26, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*  
Buckets:  
-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*  
\*\*\*\*\*

SegmentCabins:  
-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ12 2010-Jan-26, SIN-HND 2010-Jan-26, Y, 1, 0, 0, 0, 0, 200, 0,  
SQ12 2010-Jan-26, SIN-HND 2010-Jan-26, Y, 2, 0, 0, 0, 0, 200, 0,

\*\*\*\*\*  
\*\*\*\*\*  
Subclasses:  
-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ12 2010-Jan-26, SIN-HND 2010-Jan-26, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ12 2010-Jan-26, SIN-HND 2010-Jan-26, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*  
FlightDate: SQ12, 2010-Jan-27  
\*\*\*\*\*

\*\*\*\*\*  
\*\*\*\*\*  
Leg-Dates:  
-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ12 2010-Jan-27, SIN-HND, 2010-Jan-27, 09:20:00, 2010-Jan-27, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*  
\*\*\*\*\*  
LegCabins:  
-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ12 2010-Jan-27, SIN-HND 2010-Jan-27, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*  
Buckets:  
-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*  
\*\*\*\*\*

SegmentCabins:  
-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ12 2010-Jan-27, SIN-HND 2010-Jan-27, Y, 1, 0, 0, 0, 0, 200, 0,  
SQ12 2010-Jan-27, SIN-HND 2010-Jan-27, Y, 2, 0, 0, 0, 0, 200, 0,

```
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Jan-27, SIN-HND 2010-Jan-27, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Jan-27, SIN-HND 2010-Jan-27, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Jan-28
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Jan-28, SIN-HND, 2010-Jan-28, 09:20:00, 2010-Jan-28, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Jan-28, SIN-HND 2010-Jan-28, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Jan-28, SIN-HND 2010-Jan-28, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Jan-28, SIN-HND 2010-Jan-28, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Jan-28, SIN-HND 2010-Jan-28, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Jan-28, SIN-HND 2010-Jan-28, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Jan-29
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Jan-29, SIN-HND, 2010-Jan-29, 09:20:00, 2010-Jan-29, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Jan-29, SIN-HND 2010-Jan-29, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Jan-29, SIN-HND 2010-Jan-29, Y, 1, 0, 0, 0, 0, 200, 0,
```

```
SQL2 2010-Jan-29, SIN-HND 2010-Jan-29, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQL2 2010-Jan-29, SIN-HND 2010-Jan-29, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQL2 2010-Jan-29, SIN-HND 2010-Jan-29, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL2, 2010-Jan-30
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 2010-Jan-30, SIN-HND, 2010-Jan-30, 09:20:00, 2010-Jan-30, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL2 2010-Jan-30, SIN-HND 2010-Jan-30, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL2 2010-Jan-30, SIN-HND 2010-Jan-30, Y, 1, 0, 0, 0, 0, 200, 0,
SQL2 2010-Jan-30, SIN-HND 2010-Jan-30, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQL2 2010-Jan-30, SIN-HND 2010-Jan-30, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQL2 2010-Jan-30, SIN-HND 2010-Jan-30, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL2, 2010-Jan-31
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 2010-Jan-31, SIN-HND, 2010-Jan-31, 09:20:00, 2010-Jan-31, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL2 2010-Jan-31, SIN-HND 2010-Jan-31, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
```

```
SQL2 2010-Jan-31, SIN-HND 2010-Jan-31, Y, 1, 0, 0, 0, 0, 200, 0,
SQL2 2010-Jan-31, SIN-HND 2010-Jan-31, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQL2 2010-Jan-31, SIN-HND 2010-Jan-31, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQL2 2010-Jan-31, SIN-HND 2010-Jan-31, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL2, 2010-Feb-01
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 2010-Feb-01, SIN-HND, 2010-Feb-01, 09:20:00, 2010-Feb-01, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL2 2010-Feb-01, SIN-HND 2010-Feb-01, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL2 2010-Feb-01, SIN-HND 2010-Feb-01, Y, 1, 0, 0, 0, 0, 200, 0,
SQL2 2010-Feb-01, SIN-HND 2010-Feb-01, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQL2 2010-Feb-01, SIN-HND 2010-Feb-01, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQL2 2010-Feb-01, SIN-HND 2010-Feb-01, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL2, 2010-Feb-02
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 2010-Feb-02, SIN-HND, 2010-Feb-02, 09:20:00, 2010-Feb-02, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL2 2010-Feb-02, SIN-HND 2010-Feb-02, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
```

```
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-02, SIN-HND 2010-Feb-02, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-02, SIN-HND 2010-Feb-02, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-02, SIN-HND 2010-Feb-02, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-02, SIN-HND 2010-Feb-02, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-03
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-03, SIN-HND, 2010-Feb-03, 09:20:00, 2010-Feb-03, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-03, SIN-HND 2010-Feb-03, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-03, SIN-HND 2010-Feb-03, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-03, SIN-HND 2010-Feb-03, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-03, SIN-HND 2010-Feb-03, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-03, SIN-HND 2010-Feb-03, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-04
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-04, SIN-HND, 2010-Feb-04, 09:20:00, 2010-Feb-04, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-04, SIN-HND 2010-Feb-04, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
```

```
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-04, SIN-HND 2010-Feb-04, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-04, SIN-HND 2010-Feb-04, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-04, SIN-HND 2010-Feb-04, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-04, SIN-HND 2010-Feb-04, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-05
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-05, SIN-HND, 2010-Feb-05, 09:20:00, 2010-Feb-05, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-05, SIN-HND 2010-Feb-05, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-05, SIN-HND 2010-Feb-05, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-05, SIN-HND 2010-Feb-05, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-05, SIN-HND 2010-Feb-05, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-05, SIN-HND 2010-Feb-05, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-06
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-06, SIN-HND, 2010-Feb-06, 09:20:00, 2010-Feb-06, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-06, SIN-HND 2010-Feb-06, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
```

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQL2 2010-Feb-06, SIN-HND 2010-Feb-06, Y, 1, 0, 0, 0, 0, 200, 0,  
SQL2 2010-Feb-06, SIN-HND 2010-Feb-06, Y, 2, 0, 0, 0, 0, 200, 0,

\*\*\*\*\*

\*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQL2 2010-Feb-06, SIN-HND 2010-Feb-06, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQL2 2010-Feb-06, SIN-HND 2010-Feb-06, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

FlightDate: SQL2, 2010-Feb-07

\*\*\*\*\*

\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQL2 2010-Feb-07, SIN-HND, 2010-Feb-07, 09:20:00, 2010-Feb-07, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*

\*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQL2 2010-Feb-07, SIN-HND 2010-Feb-07, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

\*\*\*\*\*

\*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQL2 2010-Feb-07, SIN-HND 2010-Feb-07, Y, 1, 0, 0, 0, 0, 200, 0,  
SQL2 2010-Feb-07, SIN-HND 2010-Feb-07, Y, 2, 0, 0, 0, 0, 200, 0,

\*\*\*\*\*

\*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQL2 2010-Feb-07, SIN-HND 2010-Feb-07, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQL2 2010-Feb-07, SIN-HND 2010-Feb-07, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

FlightDate: SQL2, 2010-Feb-08

\*\*\*\*\*

\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQL2 2010-Feb-08, SIN-HND, 2010-Feb-08, 09:20:00, 2010-Feb-08, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*

\*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQL2 2010-Feb-08, SIN-HND 2010-Feb-08, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*

\*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

\*\*\*\*\*



\*\*\*\*\*  
SegmentCabins:  
-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ12 2010-Feb-08, SIN-HND 2010-Feb-08, Y, 1, 0, 0, 0, 0, 200, 0,  
SQ12 2010-Feb-08, SIN-HND 2010-Feb-08, Y, 2, 0, 0, 0, 0, 200, 0,

\*\*\*\*\*  
\*\*\*\*\*  
Subclasses:  
-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ12 2010-Feb-08, SIN-HND 2010-Feb-08, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ12 2010-Feb-08, SIN-HND 2010-Feb-08, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*  
FlightDate: SQ12, 2010-Feb-09  
\*\*\*\*\*  
\*\*\*\*\*

\*\*\*\*\*  
\*\*\*\*\*  
Leg-Dates:  
-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ12 2010-Feb-09, SIN-HND, 2010-Feb-09, 09:20:00, 2010-Feb-09, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*  
\*\*\*\*\*  
LegCabins:  
-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ12 2010-Feb-09, SIN-HND 2010-Feb-09, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*  
Buckets:  
-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*  
\*\*\*\*\*  
SegmentCabins:  
-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ12 2010-Feb-09, SIN-HND 2010-Feb-09, Y, 1, 0, 0, 0, 0, 200, 0,  
SQ12 2010-Feb-09, SIN-HND 2010-Feb-09, Y, 2, 0, 0, 0, 0, 200, 0,

\*\*\*\*\*  
\*\*\*\*\*  
Subclasses:  
-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ12 2010-Feb-09, SIN-HND 2010-Feb-09, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ12 2010-Feb-09, SIN-HND 2010-Feb-09, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*  
FlightDate: SQ12, 2010-Feb-10  
\*\*\*\*\*  
\*\*\*\*\*

\*\*\*\*\*  
\*\*\*\*\*  
Leg-Dates:  
-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ12 2010-Feb-10, SIN-HND, 2010-Feb-10, 09:20:00, 2010-Feb-10, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

\*\*\*\*\*  
\*\*\*\*\*  
LegCabins:  
-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ12 2010-Feb-10, SIN-HND 2010-Feb-10, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

\*\*\*\*\*  
\*\*\*\*\*  
Buckets:  
-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

```
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-10, SIN-HND 2010-Feb-10, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-10, SIN-HND 2010-Feb-10, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-10, SIN-HND 2010-Feb-10, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-10, SIN-HND 2010-Feb-10, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-11
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-11, SIN-HND, 2010-Feb-11, 09:20:00, 2010-Feb-11, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-11, SIN-HND 2010-Feb-11, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-11, SIN-HND 2010-Feb-11, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-11, SIN-HND 2010-Feb-11, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-11, SIN-HND 2010-Feb-11, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-11, SIN-HND 2010-Feb-11, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-12
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-12, SIN-HND, 2010-Feb-12, 09:20:00, 2010-Feb-12, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-12, SIN-HND 2010-Feb-12, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
```

```
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-12, SIN-HND 2010-Feb-12, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-12, SIN-HND 2010-Feb-12, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-12, SIN-HND 2010-Feb-12, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-12, SIN-HND 2010-Feb-12, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-13
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-13, SIN-HND, 2010-Feb-13, 09:20:00, 2010-Feb-13, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-13, SIN-HND 2010-Feb-13, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-13, SIN-HND 2010-Feb-13, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-13, SIN-HND 2010-Feb-13, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-13, SIN-HND 2010-Feb-13, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-13, SIN-HND 2010-Feb-13, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-14
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-14, SIN-HND, 2010-Feb-14, 09:20:00, 2010-Feb-14, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-14, SIN-HND 2010-Feb-14, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
```

```
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-14, SIN-HND 2010-Feb-14, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-14, SIN-HND 2010-Feb-14, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-14, SIN-HND 2010-Feb-14, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-14, SIN-HND 2010-Feb-14, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-15
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-15, SIN-HND, 2010-Feb-15, 09:20:00, 2010-Feb-15, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-15, SIN-HND 2010-Feb-15, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-15, SIN-HND 2010-Feb-15, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-15, SIN-HND 2010-Feb-15, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-15, SIN-HND 2010-Feb-15, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-15, SIN-HND 2010-Feb-15, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-16
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-16, SIN-HND, 2010-Feb-16, 09:20:00, 2010-Feb-16, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-16, SIN-HND 2010-Feb-16, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
```

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*  
\*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ12 2010-Feb-16, SIN-HND 2010-Feb-16, Y, 1, 0, 0, 0, 0, 200, 0,  
SQ12 2010-Feb-16, SIN-HND 2010-Feb-16, Y, 2, 0, 0, 0, 0, 200, 0,  
\*\*\*\*\*  
\*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ12 2010-Feb-16, SIN-HND 2010-Feb-16, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ12 2010-Feb-16, SIN-HND 2010-Feb-16, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
\*\*\*\*\*  
\*\*\*\*\*

FlightDate: SQ12, 2010-Feb-17

\*\*\*\*\*  
\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ12 2010-Feb-17, SIN-HND, 2010-Feb-17, 09:20:00, 2010-Feb-17, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,  
\*\*\*\*\*  
\*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ12 2010-Feb-17, SIN-HND 2010-Feb-17, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,  
\*\*\*\*\*  
\*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*  
\*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ12 2010-Feb-17, SIN-HND 2010-Feb-17, Y, 1, 0, 0, 0, 0, 200, 0,  
SQ12 2010-Feb-17, SIN-HND 2010-Feb-17, Y, 2, 0, 0, 0, 0, 200, 0,  
\*\*\*\*\*  
\*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ12 2010-Feb-17, SIN-HND 2010-Feb-17, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ12 2010-Feb-17, SIN-HND 2010-Feb-17, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
\*\*\*\*\*  
\*\*\*\*\*

FlightDate: SQ12, 2010-Feb-18

\*\*\*\*\*  
\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ12 2010-Feb-18, SIN-HND, 2010-Feb-18, 09:20:00, 2010-Feb-18, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,  
\*\*\*\*\*  
\*\*\*\*\*

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ12 2010-Feb-18, SIN-HND 2010-Feb-18, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,  
\*\*\*\*\*

\*\*\*\*\*  
Buckets:  
-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*  
\*\*\*\*\*  
SegmentCabins:  
-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ12 2010-Feb-18, SIN-HND 2010-Feb-18, Y, 1, 0, 0, 0, 0, 200, 0,  
SQ12 2010-Feb-18, SIN-HND 2010-Feb-18, Y, 2, 0, 0, 0, 0, 200, 0,  
\*\*\*\*\*  
\*\*\*\*\*  
Subclasses:  
-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ12 2010-Feb-18, SIN-HND 2010-Feb-18, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ12 2010-Feb-18, SIN-HND 2010-Feb-18, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
\*\*\*\*\*  
\*\*\*\*\*

FlightDate: SQ12, 2010-Feb-19  
\*\*\*\*\*  
\*\*\*\*\*  
Leg-Dates:  
-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ12 2010-Feb-19, SIN-HND, 2010-Feb-19, 09:20:00, 2010-Feb-19, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,  
\*\*\*\*\*  
\*\*\*\*\*

LegCabins:  
-----  
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ12 2010-Feb-19, SIN-HND 2010-Feb-19, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,  
\*\*\*\*\*  
\*\*\*\*\*

Buckets:  
-----  
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
\*\*\*\*\*  
\*\*\*\*\*  
SegmentCabins:  
-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
SQ12 2010-Feb-19, SIN-HND 2010-Feb-19, Y, 1, 0, 0, 0, 0, 200, 0,  
SQ12 2010-Feb-19, SIN-HND 2010-Feb-19, Y, 2, 0, 0, 0, 0, 200, 0,  
\*\*\*\*\*  
\*\*\*\*\*  
Subclasses:  
-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
SQ12 2010-Feb-19, SIN-HND 2010-Feb-19, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ12 2010-Feb-19, SIN-HND 2010-Feb-19, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
\*\*\*\*\*  
\*\*\*\*\*

FlightDate: SQ12, 2010-Feb-20  
\*\*\*\*\*  
\*\*\*\*\*  
Leg-Dates:  
-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ12 2010-Feb-20, SIN-HND, 2010-Feb-20, 09:20:00, 2010-Feb-20, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,  
\*\*\*\*\*  
\*\*\*\*\*

LegCabins:  
-----  
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
SQ12 2010-Feb-20, SIN-HND 2010-Feb-20, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

```
*****
*****
Buckets:
-----
```

```
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
```

```
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-20, SIN-HND 2010-Feb-20, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-20, SIN-HND 2010-Feb-20, Y, 2, 0, 0, 0, 0, 200, 0,
```

```
*****
*****
Subclasses:
-----
```

```
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-20, SIN-HND 2010-Feb-20, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-20, SIN-HND 2010-Feb-20, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
```

```
*****
*****
FlightDate: SQ12, 2010-Feb-21
*****
*****
```

```
Leg-Dates:
-----
```

```
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-21, SIN-HND, 2010-Feb-21, 09:20:00, 2010-Feb-21, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
```

```
LegCabins:
-----
```

```
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-21, SIN-HND 2010-Feb-21, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
```

```
*****
*****
Buckets:
-----
```

```
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
```

```
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-21, SIN-HND 2010-Feb-21, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-21, SIN-HND 2010-Feb-21, Y, 2, 0, 0, 0, 0, 200, 0,
```

```
*****
*****
Subclasses:
-----
```

```
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-21, SIN-HND 2010-Feb-21, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-21, SIN-HND 2010-Feb-21, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
```

```
*****
*****
FlightDate: SQ12, 2010-Feb-22
*****
*****
```

```
Leg-Dates:
-----
```

```
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-22, SIN-HND, 2010-Feb-22, 09:20:00, 2010-Feb-22, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
```

```
LegCabins:
-----
```

```
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
```

```
SQL2 2010-Feb-22, SIN-HND 2010-Feb-22, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL2 2010-Feb-22, SIN-HND 2010-Feb-22, Y, 1, 0, 0, 0, 0, 200, 0,
SQL2 2010-Feb-22, SIN-HND 2010-Feb-22, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQL2 2010-Feb-22, SIN-HND 2010-Feb-22, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQL2 2010-Feb-22, SIN-HND 2010-Feb-22, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL2, 2010-Feb-23
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 2010-Feb-23, SIN-HND, 2010-Feb-23, 09:20:00, 2010-Feb-23, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL2 2010-Feb-23, SIN-HND 2010-Feb-23, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL2 2010-Feb-23, SIN-HND 2010-Feb-23, Y, 1, 0, 0, 0, 0, 200, 0,
SQL2 2010-Feb-23, SIN-HND 2010-Feb-23, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQL2 2010-Feb-23, SIN-HND 2010-Feb-23, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQL2 2010-Feb-23, SIN-HND 2010-Feb-23, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL2, 2010-Feb-24
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 2010-Feb-24, SIN-HND, 2010-Feb-24, 09:20:00, 2010-Feb-24, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
```



```
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-24, SIN-HND 2010-Feb-24, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-24, SIN-HND 2010-Feb-24, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-24, SIN-HND 2010-Feb-24, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-24, SIN-HND 2010-Feb-24, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-24, SIN-HND 2010-Feb-24, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-25
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-25, SIN-HND, 2010-Feb-25, 09:20:00, 2010-Feb-25, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-25, SIN-HND 2010-Feb-25, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-25, SIN-HND 2010-Feb-25, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-25, SIN-HND 2010-Feb-25, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-25, SIN-HND 2010-Feb-25, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-25, SIN-HND 2010-Feb-25, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-26
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-26, SIN-HND, 2010-Feb-26, 09:20:00, 2010-Feb-26, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
```

```
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-26, SIN-HND 2010-Feb-26, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-26, SIN-HND 2010-Feb-26, Y, 1, 0, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-26, SIN-HND 2010-Feb-26, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-26, SIN-HND 2010-Feb-26, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-26, SIN-HND 2010-Feb-26, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-27
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-27, SIN-HND, 2010-Feb-27, 09:20:00, 2010-Feb-27, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-27, SIN-HND 2010-Feb-27, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-27, SIN-HND 2010-Feb-27, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-27, SIN-HND 2010-Feb-27, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-27, SIN-HND 2010-Feb-27, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-27, SIN-HND 2010-Feb-27, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-28
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-28, SIN-HND, 2010-Feb-28, 09:20:00, 2010-Feb-28, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
```

LegCabins:

-----

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,  
 SQL2 2010-Feb-28, SIN-HND 2010-Feb-28, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,  
 \*\*\*\*\*

Buckets:

-----

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,  
 \*\*\*\*\*

SegmentCabins:

-----

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,  
 SQL2 2010-Feb-28, SIN-HND 2010-Feb-28, Y, 1, 0, 0, 0, 0, 200, 0,  
 SQL2 2010-Feb-28, SIN-HND 2010-Feb-28, Y, 2, 0, 0, 0, 0, 200, 0,  
 \*\*\*\*\*

Subclasses:

-----

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,  
 SQL2 2010-Feb-28, SIN-HND 2010-Feb-28, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
 SQL2 2010-Feb-28, SIN-HND 2010-Feb-28, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
 \*\*\*\*\*

### 12.12.6 Exploring the Predefined BOM Tree

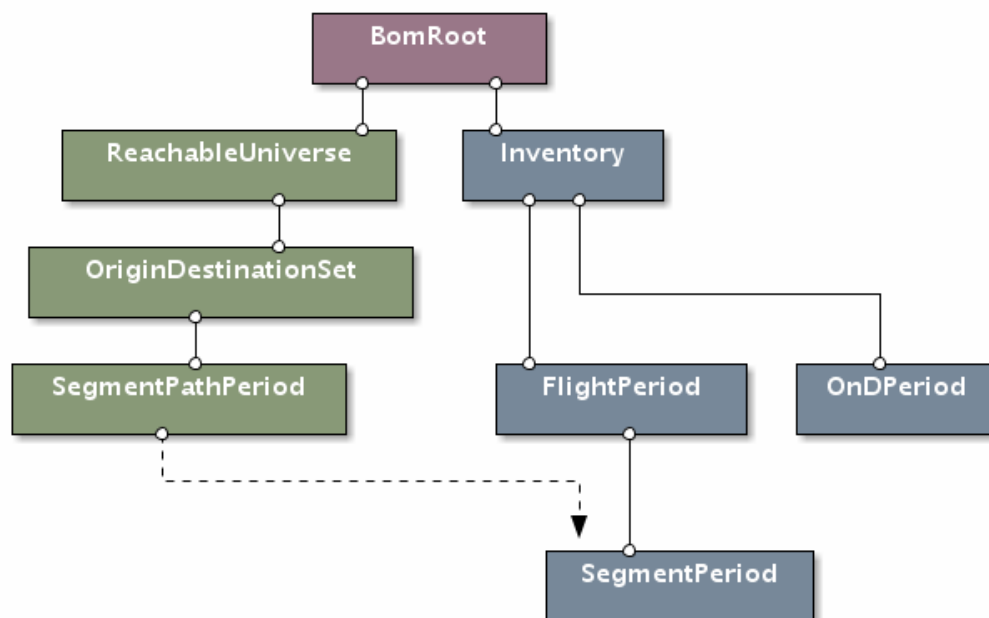


Figure 2: AirTSP BOM tree

AirTSP predefines a BOM (Business Object Model) tree specific to the airline IT arena.

#### 12.12.6.1 Airline Network BOM Tree

- [AIRTSP::ReachableUniverse](#)
- [AIRTSP::OriginDestinationSet](#)
- [AIRTSP::SegmentPathPeriod](#)

#### 12.12.6.2 Airline Schedule BOM Tree

- `stdair::Inventory`
- `stdair::FlightPeriod`
- `stdair::SegmentPeriod`
- `stdair::OnDPeriod`

#### 12.12.7 Extending the BOM Tree

#### 12.12.8 The travel solution calculation procedure

The project AirTSP aims at calculating a list of travel solutions for every incoming booking request.

### 12.13 Supported Systems

#### 12.13.1 Table of Contents

- [Introduction](#)
- [AirTSP 0.2.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 AirTSP 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 AirTSP External](#)
    - \* [Microsoft Windows XP with MS Visual C++ and Intel MKL](#)
  - [Unix Systems](#)
    - \* [SunOS 5.9 with AirTSP External](#)

- [AirTSP 3.9.1](#)
- [AirTSP 3.9.0](#)
- [AirTSP 3.8.1](#)

### 12.13.2 Introduction

This page is intended to provide a list of AirTSP supported systems, i.e. the systems on which configuration, installation and testing process of the AirTSP 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 AirTSP library on a system not mentioned below, please let us know, so we could update this database.

### 12.13.3 AirTSP 0.2.x

#### 12.13.3.1 Linux Systems

##### Fedora Core 4 with ATLAS

- **Platform:** Intel Pentium 4
- **Operating System:** Fedora Core 4 (x86)
- **Compiler:** g++ (GCC) 4.0.2 20051125
- **AirTSP release:** 0.2.0
- **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:** AirTSP 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
- **AirTSP release:** 0.2.1
- **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
```

AirTSP 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
- **AirTSP release:** 0.2.1
- **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
```

AirTSP 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
- **AirTSP release:** 0.2.0
- **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:** AirTSP 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
- **AirTSP release:** 0.2.1
- **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
```

AirTSP configured with:

```
% ./configure --with-blas="-lblas"
```

- **Date:** March 31, 2006
- **Tester:** Adam Piatyszek (ediap)

**Red Hat Enterprise Linux with AirTSP 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)
- **AirTSP release:** 0.2.0
- **External Libraries:** BLAS, CBLAS, LAPACK and FFTW libraries from AirTSP 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
- **AirTSP release:** 0.2.0
- **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:** AirTSP 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
- **AirTSP release:** 0.2.0
- **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:** AirTSP 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)

**12.13.3.2 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)
- **AirTSP release:** 0.2.1
- **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. AirTSP 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)
- **AirTSP release:** 0.2.1
- **External Libraries:** Installed from Cygwin's repository:

```
- fftw-3.0.1-2  
- fftw-dev-3.0.1-1
```

ATLAS BLAS and LAPACK libraries from AirTSP External 2.1.1 package configured using:

```
% ./configure --enable-atlas --disable-fftw
```

- **Tests Status:** All tests PASSED
- **Comments:** Only static library can be built. AirTSP 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)
- **AirTSP release:** 0.2.2
- **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. AirTSP 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)
- **AirTSP release:** 0.2.2
- **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. AirTSP 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 AirTSP 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)
- **AirTSP release:** 0.2.5
- **External Libraries:** BLAS, CBLAS, LAPACK and FFTW libraries from AirTSP External 2.2.0 package
- **Tests Status:** All tests PASSED
- **Comments:** Only static library can be built. AirTSP 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
- **AirTSP release:** 0.2.5
- **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 AirTSP based programs compiled and run with success.
- **Comments:** Only static library can be built. AirTSP built by opening the "win32\airtsp.vcproj" project file in MSVC++ and executing "Build → Build Solution" command from menu.
- **Date:** August 11, 2006
- **Tester:** Adam Piatyszek (ediap)

**12.13.3.3 Unix Systems****SunOS 5.9 with AirTSP External**

- **Platform:** SUNW, Sun-Blade-100 (SPARC)
- **Operating System:** SunOS 5.9 Generic\_112233-10
- **Compiler(s):** g++ (GCC) 3.4.5
- **AirTSP release:** 0.2.2
- **External Libraries:** BLAS, CBLAS, LAPACK and FFTW libraries from AirTSP 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:** AirTSP 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)

## 12.14 AirTSP Supported Systems (Previous Releases)

### 12.14.1 AirTSP 3.9.1

### 12.14.2 AirTSP 3.9.0

### 12.14.3 AirTSP 3.8.1

## 12.15 Tutorials

### 12.15.1 Table of Contents

- [Preparing the AirTSP Project for Development](#)
- [Your first networkBuilde](#)
  - [Summary of the different steps](#)
  - [Result of the Batch Program](#)
- [Network building with an input file](#)
  - [How to build a network input file?](#)
  - [Building the BOM tree with an input file](#)
  - [Result of the Batch Program](#)

### 12.15.2 Preparing the AirTSP Project for Development

The source code for these examples can be found in the `batches` and `test/airtsp` directories. They are compiled along with the rest of the AirTSP project. See the [Users Guide](#) for more details on how to build the AirTSP project.

### 12.15.3 Your first networkBuilde

**12.15.3.1 Summary of the different steps** All the steps below can be found in the same order in the batch `AirTSP.cpp` program.

First, we instanciate the `AIRTSP_Service` object:

Then, we construct a default sample list of travel solutions and a default booking request (as mentionned in `ug_procedure_bookingrequest` and `ug_procedure_travelsolution` parts):

```
stdair::TravelSolutionList_T lTravelSolutionList;  
airtspService.buildSegmentPathList (lTravelSolutionList, lBookingRequest);
```

For basic use, the default BOM tree can be built using:

The main step is the network building (see [The travel solution calculation procedure](#)):

**12.15.3.2 Result of the Batch Program** When the `AirTSP.cpp` program is run (with the `-b` option), the log output file should look like:

What is interesting is to compare the travel solution list (here reduced to a single travel solution) displayed before:

and after the network building:

Between the two groups of dashes, we can see that a network option structure has been added by the network builder: the price is 450 EUR for the Y class, the ticket is refundable but there are exchange fees and the customer must stay over on saturday night.

Let's return to our default BOM tree display: the only network rule stored was a match for the travel solution into consideration (same origin airport, same destination airport, flight date included in the network rule date range, same airline "BA", ...).

By looking at the network rule trip type "RT", we can guess we face a round trip network: that means the price given in the default bom tree construction in `stdair::CmdBomManager.hpp` has been divided by 2 because we are considering either an inbound trip or an outbound one.

## 12.15.4 Network building with an input file

**12.15.4.1 How to build a network input file?** The objective here is to build a network input file to network build the default travel solution list built using:

This travel solution list, reduced to a singleton, can be displayed as done before:

We deduce:

- we need a network rule whose origin-destination couple is "LHR, SYD".
- the date range must include the date "2011-06-10".
- the time range must include the time "21:45".
- the airline operating is "BA", so it must be the airline pricing.

We can deduce a part of our network rule file :

```
// Fares: fare ID; OriginCity; DestinationCity; TripType; DateRangeStart; DateRangeEnd; DepartureTimeRange
// Segment: AirlineCode; Class;
1; LHR; SYD; ??; 2011-01-01; 2011-12-31; 00:00; 23:59; ???; ?; ??; ?; ?; ?; ?; ?; ?; ?; BA; ?;
```

We have no information about stay duration and advance purchase (such information are contained into the booking request): so let us put "0" to embrace all the requests possible.

No information for the point-of-sale and the channel too: let us consider all the channels ("IN", "DN", "IF" and "DF") and all the points of sale (the origin "LHR", the destination "SYD" and the rest-of-the-world "ROW") existing. To access this information, we could look into the default booking request.

The input file is now:

```
// Fares: fare ID; OriginCity; DestinationCity; TripType; DateRangeStart; DateRangeEnd; DepartureTimeRange
// Segment: AirlineCode; Class;
1; LHR; SYD; ??; 2011-01-01; 2011-12-31; 00:00; 23:59; LHR; ?; IN; 0; ?; ?; ?; 0; ????; BA; ?;
2; LHR; SYD; ??; 2011-01-01; 2011-12-31; 00:00; 23:59; LHR; ?; IF; 0; ?; ?; ?; 0; ????; BA; ?;
3; LHR; SYD; ??; 2011-01-01; 2011-12-31; 00:00; 23:59; LHR; ?; DN; 0; ?; ?; ?; 0; ????; BA; ?;
4; LHR; SYD; ??; 2011-01-01; 2011-12-31; 00:00; 23:59; LHR; ?; DF; 0; ?; ?; ?; 0; ????; BA; ?;
5; LHR; SYD; ??; 2011-01-01; 2011-12-31; 00:00; 23:59; SYD; ?; IN; 0; ?; ?; ?; 0; ????; BA; ?;
6; LHR; SYD; ??; 2011-01-01; 2011-12-31; 00:00; 23:59; SYD; ?; IF; 0; ?; ?; ?; 0; ????; BA; ?;
7; LHR; SYD; ??; 2011-01-01; 2011-12-31; 00:00; 23:59; SYD; ?; DN; 0; ?; ?; ?; 0; ????; BA; ?;
8; LHR; SYD; ??; 2011-01-01; 2011-12-31; 00:00; 23:59; SYD; ?; DF; 0; ?; ?; ?; 0; ????; BA; ?;
9; LHR; SYD; ??; 2011-01-01; 2011-12-31; 00:00; 23:59; ROW; ?; IN; 0; ?; ?; ?; 0; ????; BA; ?;
10; LHR; SYD; ??; 2011-01-01; 2011-12-31; 00:00; 23:59; ROW; ?; IF; 0; ?; ?; ?; 0; ????; BA; ?;
11; LHR; SYD; ??; 2011-01-01; 2011-12-31; 00:00; 23:59; ROW; ?; DN; 0; ?; ?; ?; 0; ????; BA; ?;
12; LHR; SYD; ??; 2011-01-01; 2011-12-31; 00:00; 23:59; ROW; ?; DF; 0; ?; ?; ?; 0; ????; BA; ?;
```

Let us say we have just the Economy cabin "Y" and British Airways prices ticket for class "Y".

No information about the trip type, so we duplicate all the network rules for both type: one-way "OW" and round-trip "RT" (to access this information, we could look to the default booking request).

The network options are all set to a default value "T" (meaning true) and the network values are chosen to be all distinct.

We obtain:

```
// Fares: fare ID; OriginCity; DestinationCity; TripType; DateRangeStart; DateRangeEnd; DepartureTimeRange
// Segment: AirlineCode; Class;
1; LHR; SYD; OW; 2011-01-01; 2011-12-31; 00:00; 23:59; LHR; Y; IN; 0; T; T; T; 0; 50; BA; Y;
2; LHR; SYD; OW; 2011-01-01; 2011-12-31; 00:00; 23:59; LHR; Y; IF; 0; T; T; T; 0; 150; BA; Y;
3; LHR; SYD; OW; 2011-01-01; 2011-12-31; 00:00; 23:59; LHR; Y; DN; 0; T; T; T; 0; 250; BA; Y;
4; LHR; SYD; OW; 2011-01-01; 2011-12-31; 00:00; 23:59; LHR; Y; DF; 0; T; T; T; 0; 350; BA; Y;
5; LHR; SYD; OW; 2011-01-01; 2011-12-31; 00:00; 23:59; SYD; Y; IN; 0; T; T; T; 0; 450; BA; Y;
6; LHR; SYD; OW; 2011-01-01; 2011-12-31; 00:00; 23:59; SYD; Y; IF; 0; T; T; T; 0; 550; BA; Y;
7; LHR; SYD; OW; 2011-01-01; 2011-12-31; 00:00; 23:59; SYD; Y; DN; 0; T; T; T; 0; 650; BA; Y;
8; LHR; SYD; OW; 2011-01-01; 2011-12-31; 00:00; 23:59; SYD; Y; DF; 0; T; T; T; 0; 750; BA; Y;
9; LHR; SYD; OW; 2011-01-01; 2011-12-31; 00:00; 23:59; ROW; Y; IN; 0; T; T; T; 0; 850; BA; Y;
10; LHR; SYD; OW; 2011-01-01; 2011-12-31; 00:00; 23:59; ROW; Y; IF; 0; T; T; T; 0; 950; BA; Y;
11; LHR; SYD; OW; 2011-01-01; 2011-12-31; 00:00; 23:59; ROW; Y; DN; 0; T; T; T; 0; 1050; BA; Y;
12; LHR; SYD; OW; 2011-01-01; 2011-12-31; 00:00; 23:59; ROW; Y; DF; 0; T; T; T; 0; 1150; BA; Y;
13; LHR; SYD; RT; 2011-01-01; 2011-12-31; 00:00; 23:59; LHR; Y; IN; 0; T; T; T; 0; 90; BA; Y;
14; LHR; SYD; RT; 2011-01-01; 2011-12-31; 00:00; 23:59; LHR; Y; IF; 0; T; T; T; 0; 190; BA; Y;
15; LHR; SYD; RT; 2011-01-01; 2011-12-31; 00:00; 23:59; LHR; Y; DN; 0; T; T; T; 0; 290; BA; Y;
16; LHR; SYD; RT; 2011-01-01; 2011-12-31; 00:00; 23:59; LHR; Y; DF; 0; T; T; T; 0; 390; BA; Y;
17; LHR; SYD; RT; 2011-01-01; 2011-12-31; 00:00; 23:59; SYD; Y; IN; 0; T; T; T; 0; 490; BA; Y;
18; LHR; SYD; RT; 2011-01-01; 2011-12-31; 00:00; 23:59; SYD; Y; IF; 0; T; T; T; 0; 590; BA; Y;
19; LHR; SYD; RT; 2011-01-01; 2011-12-31; 00:00; 23:59; SYD; Y; DN; 0; T; T; T; 0; 690; BA; Y;
20; LHR; SYD; RT; 2011-01-01; 2011-12-31; 00:00; 23:59; SYD; Y; DF; 0; T; T; T; 0; 790; BA; Y;
21; LHR; SYD; RT; 2011-01-01; 2011-12-31; 00:00; 23:59; ROW; Y; IN; 0; T; T; T; 0; 890; BA; Y;
22; LHR; SYD; RT; 2011-01-01; 2011-12-31; 00:00; 23:59; ROW; Y; IF; 0; T; T; T; 0; 990; BA; Y;
23; LHR; SYD; RT; 2011-01-01; 2011-12-31; 00:00; 23:59; ROW; Y; DN; 0; T; T; T; 0; 1090; BA; Y;
24; LHR; SYD; RT; 2011-01-01; 2011-12-31; 00:00; 23:59; ROW; Y; DF; 0; T; T; T; 0; 1190; BA; Y;
```

**12.15.4.2 Building the BOM tree with an input file** The steps are the same as before [Summary of the different steps](#) except the bom tree must be built using the network input file :

**12.15.4.3 Result of the Batch Program** When the `AirTSP.cpp` program is run with the `-f` option linking with the file built just above:

```
~/AirTSP -f ~/<YourFileName>.csv
```

the last lines of the log output should look like:

```
[D]~/AirTSPgit/AirTSP/batches/AirTSP.cpp:223: Travel solutions:
  [0] [0] BA, 9, 2011-06-10, LHR, SYD, 21:45 --- Y, 145, 1 1 1 ---
```

We have just one network option added to the travel solution. We can deduce from the price value 145 that the network builder used the network rule number 15 to price the travel solution. We have an inbound or outbound trip of a round trip: the total price 290 has been divided by 2.

## 12.16 Command-Line Test to Demonstrate How To Test the AirTSP Project

```
*/
// //////////////////////////////////////
// Import section
// //////////////////////////////////////
// STL
#include <sstream>
#include <fstream>
#include <string>
// Boost Unit Test Framework (UTF)
#define BOOST_TEST_DYN_LINK
#define BOOST_TEST_MAIN
#define BOOST_TEST_MODULE InventoryTestSuite
#include <boost/test/unit_test.hpp>
// Boost Date-Time
#include <boost/date_time/gregorian/gregorian.hpp>
// StdAir
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/basic/BasLogParams.hpp>
#include <stdair/basic/BasDBParams.hpp>
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/bom/TravelSolutionStruct.hpp>
#include <stdair/bom/BookingRequestStruct.hpp>
#include <stdair/service/Logger.hpp>
// AirTSP
#include <airtsp/AIRTSP_Types.hpp>
#include <airtsp/AIRTSP_Service.hpp>
#include <airtsp/config/airtsp-paths.hpp>

namespace boost_utf = boost::unit_test;

// (Boost) Unit Test XML Report
std::ofstream utfReportStream ("AirlineScheduleTestSuite_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() {
    }
}
```



```

};

// //////////////////////////////////////
const unsigned int testScheduleHelper (const unsigned short iTestFlag,
                                       const stdair::Filename_T& iScheduleInputFilename,
                                       const stdair::Filename_T& iODInputFilename,
                                       const bool isBuiltin,
                                       const bool isWithOnD) {

    // Output log File
    std::ostringstream oStr;
    oStr << "AirlineScheduleTestSuite_" << 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();

    // Instantiate the AirTSP service
    const stdair::BasLogParams lLogParams (stdair::LOG::DEBUG, logOutputFile);
    AIRTSP::AIRTSP_Service airtspService (lLogParams);

    stdair::AirportCode_T lOrigin;
    stdair::AirportCode_T lDestination;
    stdair::AirportCode_T lPOS;
    stdair::Date_T lPreferredDepartureDate;;
    stdair::Date_T lRequestDate;

    // Check whether or not a (CSV) input file should be read
    if (isBuiltin == true) {

        // Build the default sample BOM tree (filled with schedules)
        airtspService.buildSampleBom();

        lOrigin = "SIN";
        lDestination = "BKK";
        lPOS = "SIN";
        lPreferredDepartureDate = boost::gregorian::from_string ("2010/02/08");
        lRequestDate = boost::gregorian::from_string ("2010/01/21");

    } else {

        if (isWithOnD == false) {

            // Build the BOM tree from parsing input files
            const stdair::ScheduleFilePath lScheduleFilePath (iScheduleInputFilename);
            airtspService.parseAndLoad (lScheduleFilePath);

            lOrigin = "NCE";
            lDestination = "BKK";
            lPOS = "NCE";
            lPreferredDepartureDate = boost::gregorian::from_string ("2007/04/21");
            lRequestDate = boost::gregorian::from_string ("2007/03/21");

        } else {

            // Build the BOM tree from parsing input files
            const stdair::ScheduleFilePath lScheduleFilePath (iScheduleInputFilename);
            const stdair::ODFilePath lODFilePath (iODInputFilename);
            airtspService.parseAndLoad (lScheduleFilePath,
                                       lODFilePath);

            lOrigin = "SIN";
            lDestination = "BKK";
            lPOS = "SIN";

```

```

        lPreferredDepartureDate = boost::gregorian::from_string ("2012/06/04");
        lRequestDate = boost::gregorian::from_string ("2012/01/01");
    }

}

// Create a booking request structure
const stdair::Duration_T lRequestTime (boost::posix_time::hours(8));
const stdair::DateTime_T lRequestDateTime (lRequestDate, lRequestTime);
const stdair::CabinCode_T lPreferredCabin ("Bus");
const stdair::PartySize_T lPartySize (3);
const stdair::ChannelLabel_T lChannel ("DF");
const stdair::TripType_T lTripType ("RO");
const stdair::DayDuration_T lStayDuration (5);
const stdair::FrequentFlyer_T lFrequentFlyerType ("NONE");
const stdair::Duration_T lPreferredDepartureTime (boost::posix_time::hours(10));
const stdair::WTP_T lWTP (2000.0);
const stdair::PriceValue_T lValueOfTime (20.0);
const stdair::ChangeFees_T lChangeFees (true);
const stdair::Disutility_T lChangeFeeDisutility (50);
const stdair::NonRefundable_T lNonRefundable (true);
const stdair::Disutility_T lNonRefundableDisutility (50);

const stdair::BookingRequestStruct lBookingRequest (lOrigin, lDestination,
                                                    lPOS,
                                                    lPreferredDepartureDate,
                                                    lRequestDateTime,
                                                    lPreferredCabin,
                                                    lPartySize, lChannel,
                                                    lTripType, lStayDuration,
                                                    lFrequentFlyerType,
                                                    lPreferredDepartureTime,
                                                    lWTP, lValueOfTime,
                                                    lChangeFees,
                                                    lChangeFeeDisutility,
                                                    lNonRefundable,
                                                    lNonRefundableDisutility);

// Build the segment path list
stdair::TravelSolutionList_T lTravelSolutionList;
airtspService.buildSegmentPathList (lTravelSolutionList, lBookingRequest);
const unsigned int lNbOfTravelSolutions = lTravelSolutionList.size();

STDAIR_LOG_DEBUG ("The number of travel solutions for the booking request '"
                  << lBookingRequest.describe() << "' is equal to "
                  << lNbOfTravelSolutions << ".");

// Close the Log outputFile
logOutputFile.close();

return lNbOfTravelSolutions;
}

// ////////////////////////////////// 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 (airtsp_simple_build) {

    // Input file name
    const stdair::Filename_T lScheduleInputFilename (STDAIR_SAMPLE_DIR

```

```

        "/schedule03.csv");

// State whether the BOM tree should be built-in or parsed from input files
const bool isBuiltin = false;
const bool isWithOnD = false;

// Try to build a travel solution list
unsigned int lNbOfTravelSolutions = 0;
BOOST_CHECK_NO_THROW (lNbOfTravelSolutions =
    testScheduleHelper (0, lScheduleInputFilename, " ",
        isBuiltin, isWithOnD));

// Check the size of the travel solution list
const unsigned int lExpectedNbOfTravelSolutions = 4;
BOOST_CHECK_MESSAGE (lNbOfTravelSolutions == lExpectedNbOfTravelSolutions,
    "The number of travel solutions is "
    << lNbOfTravelSolutions << ", but it should be equal to "
    << lExpectedNbOfTravelSolutions);
}

BOOST_AUTO_TEST_CASE (airtsp_default_bom_tree_simple_build) {

// State whether the BOM tree should be built-in or parsed from input files
const bool isBuiltin = true;
const bool isWithOnD = false;

// Try to build a travel solution list
unsigned int lNbOfTravelSolutions = 0;
BOOST_CHECK_NO_THROW (lNbOfTravelSolutions =
    testScheduleHelper (1, " ", " ", isBuiltin, isWithOnD));

// Check the size of the travel solution list
const unsigned int lExpectedNbOfTravelSolutions = 1;
BOOST_CHECK_MESSAGE (lNbOfTravelSolutions == lExpectedNbOfTravelSolutions,
    "The number of travel solutions is "
    << lNbOfTravelSolutions << ", but it should be equal to "
    << lExpectedNbOfTravelSolutions);
}

BOOST_AUTO_TEST_CASE (airtsp_OnD_input_file) {

// Input file names
const stdair::Filename_T lScheduleInputFilename (STDAIR_SAMPLE_DIR
    "/rds01/schedule05.csv");
const stdair::Filename_T lODInputFilename (STDAIR_SAMPLE_DIR
    "/ond01.csv");

// State whether the BOM tree should be built-in or parsed from input files
const bool isBuiltin = false;
const bool isWithOnD = true;

// Try to build a travel solution list
unsigned int lNbOfTravelSolutions = 0;
BOOST_CHECK_NO_THROW (lNbOfTravelSolutions =
    testScheduleHelper (2, lScheduleInputFilename,
        lODInputFilename,
        isBuiltin, isWithOnD));

// Check the size of the travel solution list
const unsigned int lExpectedNbOfTravelSolutions = 1;
BOOST_CHECK_MESSAGE (lNbOfTravelSolutions == lExpectedNbOfTravelSolutions,
    "The number of travel solutions is "
    << lNbOfTravelSolutions << ", but it should be equal to "
    << lExpectedNbOfTravelSolutions);
}

```

```

BOOST_AUTO_TEST_CASE (airtsp_missing_OnD_input_file) {

    // Input file names
    const stdair::Filename_T lScheduleInputFilename (STDAIR_SAMPLE_DIR
                                                    "/schedule03.csv");
    const stdair::Filename_T lODInputFilename (STDAIR_SAMPLE_DIR
                                                "/missingFiles.csv");

    // State whether the BOM tree should be built-in or parsed from input files
    const bool isBuiltin = false;
    const bool isWithOnD = true;

    // Try to build a travel solution list
    BOOST_CHECK_THROW (testScheduleHelper (3, lScheduleInputFilename,
                                           lODInputFilename,
                                           isBuiltin, isWithOnD),
                      AIRTSP::OnDInputFileNotFoundException);
}

BOOST_AUTO_TEST_CASE (airtsp_missing_schedule_input_file) {

    // Input file name
    const stdair::Filename_T lScheduleInputFilename (STDAIR_SAMPLE_DIR
                                                    "/missingFiles.csv");

    // State whether the BOM tree should be built-in or parsed from input files
    const bool isBuiltin = false;
    const bool isWithOnD = false;

    // Try to build a travel solution list
    BOOST_CHECK_THROW (testScheduleHelper (4, lScheduleInputFilename, " ",
                                           isBuiltin, isWithOnD),
                      AIRTSP::ScheduleInputFileNotFoundException);
}

BOOST_AUTO_TEST_CASE (airtsp_segment_date_not_found) {

    // Input file name
    const stdair::Filename_T lScheduleInputFilename (STDAIR_SAMPLE_DIR
                                                    "/scheduleError03.csv");

    // State whether the BOM tree should be built-in or parsed from input files
    const bool isBuiltin = false;
    const bool isWithOnD = false;

    // Try to build a travel solution list
    BOOST_CHECK_THROW (testScheduleHelper (5, lScheduleInputFilename,
                                           " ",
                                           isBuiltin, isWithOnD),
                      AIRTSP::SegmentDateNotFoundException);
}

// End the test suite
BOOST_AUTO_TEST_SUITE_END()

/*!

```