

# AirInv Reference Manual

## 1.00.0

Generated by Doxygen 1.4.7

Tue Dec 25 22:30:28 2012

## Contents

<b>1</b>	<b><a href="#">AirInv Documentation</a></b>	<b>1</b>
<b>2</b>	<b><a href="#">AirInv Directory Hierarchy</a></b>	<b>2</b>
<b>3</b>	<b><a href="#">AirInv Namespace Index</a></b>	<b>3</b>
<b>4</b>	<b><a href="#">AirInv Hierarchical Index</a></b>	<b>3</b>
<b>5</b>	<b><a href="#">AirInv Class Index</a></b>	<b>12</b>
<b>6</b>	<b><a href="#">AirInv File Index</a></b>	<b>19</b>
<b>7</b>	<b><a href="#">AirInv Page Index</a></b>	<b>24</b>
<b>8</b>	<b><a href="#">AirInv Directory Documentation</a></b>	<b>25</b>
<b>9</b>	<b><a href="#">AirInv Namespace Documentation</a></b>	<b>29</b>
<b>10</b>	<b><a href="#">AirInv Class Documentation</a></b>	<b>47</b>
<b>11</b>	<b><a href="#">AirInv File Documentation</a></b>	<b>377</b>
<b>12</b>	<b><a href="#">AirInv Page Documentation</a></b>	<b>431</b>

## 1 AirInv Documentation

### 1.1 Getting Started

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

## 1.2 AirInv at SourceForge

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

## 1.3 AirInv 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 AirInv

## 1.6 About AirInv

AirInv is a C++ library of airline inventory management classes and functions, mainly targeting simulation purposes. [N](#)

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

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

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

**Note:**

(N) - The AirInv 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 AirInv.

## 2 AirInv Directory Hierarchy

### 2.1 AirInv Directories

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

<b>airinv</b>	<b>25</b>
<b>basic</b>	<b>25</b>
<b>batches</b>	<b>26</b>
<b>bom</b>	<b>26</b>
<b>command</b>	<b>27</b>
<b>vault</b>	<b>29</b>
<b>config</b>	<b>27</b>
<b>factory</b>	<b>28</b>
<b>server</b>	<b>28</b>
<b>service</b>	<b>28</b>
<b>ui</b>	<b>29</b>
<b>cmdline</b>	<b>27</b>
<b>test</b>	<b>29</b>
<b>airinv</b>	<b>25</b>

## 3 AirInv Namespace Index

### 3.1 AirInv Namespace List

Here is a list of all namespaces with brief descriptions:

<b>AIRINV</b>	<b>29</b>
<b>AIRINV::DCPParserHelper</b>	<b>37</b>
<b>AIRINV::FFDisutilityParserHelper</b>	<b>38</b>
<b>AIRINV::FRAT5ParserHelper</b>	<b>39</b>

<a href="#">AIRINV::InventoryParserHelper</a>	39
<a href="#">AIRINV::ScheduleParserHelper</a>	44
<a href="#">stdair</a> (Forward declarations )	47

## 4 AirInv Hierarchical Index

### 4.1 AirInv Class Hierarchy

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

<a href="#">AIRINV::AIRINV_Master_Service</a>	47
<a href="#">AIRINV::AIRINV_Service</a>	57
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 >	
<a href="#">AIRINV::BomAbstract</a>	67
<a href="#">stdair::BomPropertyTree</a>	69
<a href="#">AIRINV::BomRootHelper</a>	70
<a href="#">AIRINV::BookingClassHelper</a>	71
<a href="#">enable_shared_from_this</a>	75
<a href="#">AIRINV::Connection</a>	79
<a href="#">noncopyable</a>	75
<a href="#">AIRINV::AirInvServer</a>	66
<a href="#">AIRINV::Connection</a>	79
<a href="#">AIRINV::RequestHandler</a>	202
<a href="#">grammar</a>	76
<a href="#">AIRINV::FFDisutilityParserHelper::FFDisutilityParser</a>	119
<a href="#">AIRINV::FRAT5ParserHelper::FRAT5Parser</a>	157
<a href="#">AIRINV::InventoryParserHelper::InventoryParser</a>	173
<a href="#">AIRINV::ScheduleParserHelper::FlightPeriodParser</a>	135
<a href="#">grammar</a>	76

<b>AIRINV::DCPParserHelper::DCPRuleParser</b>	<b>91</b>
<b>CmdAbstract</b>	<b>78</b>
<b>AIRINV::DCPEventGenerator</b>	<b>81</b>
<b>AIRINV::DCPParser</b>	<b>89</b>
<b>AIRINV::DCPRuleFileParser</b>	<b>90</b>
<b>AIRINV::FFDisutilityFileParser</b>	<b>116</b>
<b>AIRINV::FFDisutilityParser</b>	<b>118</b>
<b>AIRINV::FlightPeriodFileParser</b>	<b>134</b>
<b>AIRINV::FRAT5FileParser</b>	<b>154</b>
<b>AIRINV::FRAT5Parser</b>	<b>156</b>
<b>AIRINV::InventoryBuilder</b>	<b>163</b>
<b>AIRINV::InventoryFileParser</b>	<b>164</b>
<b>AIRINV::InventoryGenerator</b>	<b>166</b>
<b>AIRINV::InventoryParser</b>	<b>174</b>
<b>AIRINV::ScheduleParser</b>	<b>205</b>
std::complex	
<b>TestFixture</b>	<b>80</b>
<b>InventoryTestSuite</b>	<b>182</b>
<b>AIRINV::DefaultMap</b>	<b>95</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>AIRINV::FacBomAbstract</b>	<b>107</b>
<b>FacServiceAbstract</b>	<b>109</b>

<b>AIRINV::FacAirinvMasterServiceContext</b>	<b>104</b>
<b>AIRINV::FacAirinvServiceContext</b>	<b>105</b>
<b>AIRINV::FacServiceAbstract</b>	<b>109</b>
<b>AIRINV::FacSupervisor</b>	<b>111</b>
<b>AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition&lt; ScannerT &gt;</b>	<b>120</b>
<b>FileNotFoundException</b>	<b>124</b>
<b>AIRINV::FFDisutilityInputFileNotFoundException</b>	<b>117</b>
<b>AIRINV::FRAT5InputFileNotFoundException</b>	<b>156</b>
<b>AIRINV::InventoryInputFileNotFoundException</b>	<b>169</b>
<b>AIRINV::ScheduleInputFileNotFoundException</b>	<b>205</b>
<b>AIRINV::FlightDateHelper</b>	<b>125</b>
<b>AIRINV::ScheduleParserHelper::FlightPeriodParser::definition&lt; ScannerT &gt;</b>	<b>136</b>
<b>AIRINV::FRAT5ParserHelper::FRAT5Parser::definition&lt; ScannerT &gt;</b>	<b>158</b>
<b>AIRINV::header</b>	<b>162</b>
<b>InputFilePath</b>	<b>163</b>
<b>AIRINV::InventoryFilePath</b>	<b>165</b>
<b>AIRINV::InventoryHelper</b>	<b>167</b>
<b>AIRINV::InventoryManager</b>	<b>169</b>
<b>AIRINV::InventoryParserHelper::InventoryParser::definition&lt; ScannerT &gt;</b>	<b>175</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::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>AIRINV::LegCabinHelper</b>	<b>183</b>
std::list< T >	
std::map< K, T >	
std::multimap< K, T >	
std::multiset< K >	
<b>ObjectCreationDuplicationException</b>	<b>190</b>
<b>AIRINV::FlightDateDuplicationException</b>	<b>124</b>
<b>ObjectNotFoundException</b>	<b>191</b>
<b>AIRINV::FlightDateNotFoundException</b>	<b>126</b>
<b>AIRINV::InventoryNotFoundException</b>	<b>172</b>
<b>ParserException</b>	<b>191</b>
<b>AIRINV::SegmentDateNotFoundException</b>	<b>211</b>
<b>AIRINV::InventoryParserHelper::ParserSemanticAction</b>	<b>191</b>



<b>AIRINV::InventoryParserHelper::doEndFlightDate</b>	<b>102</b>
<b>AIRINV::InventoryParserHelper::storeACP</b>	<b>216</b>
<b>AIRINV::InventoryParserHelper::storeAirlineCode</b>	<b>219</b>
<b>AIRINV::InventoryParserHelper::storeAU</b>	<b>224</b>
<b>AIRINV::InventoryParserHelper::storeBoardingDate</b>	<b>225</b>
<b>AIRINV::InventoryParserHelper::storeBoardingTime</b>	<b>229</b>
<b>AIRINV::InventoryParserHelper::storeBookingCounter</b>	<b>231</b>
<b>AIRINV::InventoryParserHelper::storeBucketAvaibility</b>	<b>232</b>
<b>AIRINV::InventoryParserHelper::storeClassAvailability</b>	<b>241</b>
<b>AIRINV::InventoryParserHelper::storeClassCode</b>	<b>243</b>
<b>AIRINV::InventoryParserHelper::storeClassETB</b>	<b>246</b>
<b>AIRINV::InventoryParserHelper::storeCumulatedProtection</b>	<b>248</b>
<b>AIRINV::InventoryParserHelper::storeETB</b>	<b>269</b>
<b>AIRINV::InventoryParserHelper::storeFamilyCode</b>	<b>273</b>
<b>AIRINV::InventoryParserHelper::storeFClasses</b>	<b>276</b>
<b>AIRINV::InventoryParserHelper::storeFlightDate</b>	<b>281</b>
<b>AIRINV::InventoryParserHelper::storeFlightNumber</b>	<b>283</b>
<b>AIRINV::InventoryParserHelper::storeFlightTypeCode</b>	<b>286</b>
<b>AIRINV::InventoryParserHelper::storeFlightVisibilityCode</b>	<b>288</b>
<b>AIRINV::InventoryParserHelper::storeGAV</b>	<b>292</b>
<b>AIRINV::InventoryParserHelper::storeLegBoardingPoint</b>	<b>294</b>
<b>AIRINV::InventoryParserHelper::storeLegCabinCode</b>	<b>299</b>
<b>AIRINV::InventoryParserHelper::storeLegOffPoint</b>	<b>302</b>
<b>AIRINV::InventoryParserHelper::storeNAV</b>	<b>305</b>
<b>AIRINV::InventoryParserHelper::storeNbOfBkgs</b>	<b>307</b>
<b>AIRINV::InventoryParserHelper::storeNbOfGroupBkgs</b>	<b>309</b>
<b>AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs</b>	<b>311</b>
<b>AIRINV::InventoryParserHelper::storeNbOfStaffBkgs</b>	<b>313</b>
<b>AIRINV::InventoryParserHelper::storeNbOfWLBkgs</b>	<b>315</b>

AIRINV::InventoryParserHelper::storeNego	317
AIRINV::InventoryParserHelper::storeNoShow	320
AIRINV::InventoryParserHelper::storeOffDate	322
AIRINV::InventoryParserHelper::storeOffTime	325
AIRINV::InventoryParserHelper::storeOperatingAirlineCode	328
AIRINV::InventoryParserHelper::storeOperatingFlightNumber	332
AIRINV::InventoryParserHelper::storeOverbooking	335
AIRINV::InventoryParserHelper::storeParentClassCode	337
AIRINV::InventoryParserHelper::storeParentSubclassCode	339
AIRINV::InventoryParserHelper::storeProtection	342
AIRINV::InventoryParserHelper::storeRevenueAvailability	344
AIRINV::InventoryParserHelper::storeSaleableCapacity	346
AIRINV::InventoryParserHelper::storeSeatIndex	349
AIRINV::InventoryParserHelper::storeSegmentAvailability	351
AIRINV::InventoryParserHelper::storeSegmentBoardingPoint	354
AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter	356
AIRINV::InventoryParserHelper::storeSegmentCabinCode	358
AIRINV::InventoryParserHelper::storeSegmentOffPoint	361
AIRINV::InventoryParserHelper::storeSnapshotDate	366
AIRINV::InventoryParserHelper::storeSubclassCode	369
AIRINV::InventoryParserHelper::storeUPR	371
AIRINV::InventoryParserHelper::storeYieldUpperRange	373
AIRINV::ScheduleParserHelper::ParserSemanticAction	193
AIRINV::ScheduleParserHelper::doEndFlight	100
AIRINV::ScheduleParserHelper::storeAirlineCode	222
AIRINV::ScheduleParserHelper::storeBoardingTime	227
AIRINV::ScheduleParserHelper::storeCapacity	236
AIRINV::ScheduleParserHelper::storeClasses	245
AIRINV::ScheduleParserHelper::storeDateRangeEnd	254

AIRINV::ScheduleParserHelper::storeDateRangeStart	256
AIRINV::ScheduleParserHelper::storeDow	263
AIRINV::ScheduleParserHelper::storeElapsedTime	266
AIRINV::ScheduleParserHelper::storeFamilyCode	271
AIRINV::ScheduleParserHelper::storeFClasses	275
AIRINV::ScheduleParserHelper::storeFFDisutilityCurveKey	278
AIRINV::ScheduleParserHelper::storeFlightNumber	284
AIRINV::ScheduleParserHelper::storeFRAT5CurveKey	290
AIRINV::ScheduleParserHelper::storeLegBoardingPoint	296
AIRINV::ScheduleParserHelper::storeLegCabinCode	298
AIRINV::ScheduleParserHelper::storeLegOffPoint	301
AIRINV::ScheduleParserHelper::storeOffTime	323
AIRINV::ScheduleParserHelper::storeOperatingAirlineCode	327
AIRINV::ScheduleParserHelper::storeOperatingFlightNumber	330
AIRINV::ScheduleParserHelper::storeSegmentBoardingPoint	352
AIRINV::ScheduleParserHelper::storeSegmentCabinCode	360
AIRINV::ScheduleParserHelper::storeSegmentOffPoint	363
AIRINV::ScheduleParserHelper::storeSegmentSpecificity	365
AIRINV::DCPParserHelper::ParserSemanticAction	195
AIRINV::DCPParserHelper::doEndDCP	98
AIRINV::DCPParserHelper::storeAdvancePurchase	217
AIRINV::DCPParserHelper::storeAirlineCode	221
AIRINV::DCPParserHelper::storeCabinCode	234
AIRINV::DCPParserHelper::storeChangeFees	237
AIRINV::DCPParserHelper::storeChannel	238
AIRINV::DCPParserHelper::storeClass	240
AIRINV::DCPParserHelper::storeDateRangeEnd	253
AIRINV::DCPParserHelper::storeDateRangeStart	257
AIRINV::DCPParserHelper::storeDCP	258

<b>AIRINV::DCPParserHelper::storeDCPIId</b>	<b>260</b>
<b>AIRINV::DCPParserHelper::storeDestination</b>	<b>261</b>
<b>AIRINV::DCPParserHelper::storeEndRangeTime</b>	<b>268</b>
<b>AIRINV::DCPParserHelper::storeMinimumStay</b>	<b>304</b>
<b>AIRINV::DCPParserHelper::storeNonRefundable</b>	<b>318</b>
<b>AIRINV::DCPParserHelper::storeOrigin</b>	<b>334</b>
<b>AIRINV::DCPParserHelper::storePOS</b>	<b>341</b>
<b>AIRINV::DCPParserHelper::storeSaturdayStay</b>	<b>347</b>
<b>AIRINV::DCPParserHelper::storeStartRangeTime</b>	<b>368</b>
<b>AIRINV::FFDisutilityParserHelper::ParserSemanticAction</b>	<b>197</b>
<b>AIRINV::FFDisutilityParserHelper::doEndCurve</b>	<b>96</b>
<b>AIRINV::FFDisutilityParserHelper::storeCurveKey</b>	<b>250</b>
<b>AIRINV::FFDisutilityParserHelper::storeDTD</b>	<b>264</b>
<b>AIRINV::FFDisutilityParserHelper::storeFFDisutilityValue</b>	<b>280</b>
<b>AIRINV::FRAT5ParserHelper::ParserSemanticAction</b>	<b>198</b>
<b>AIRINV::FRAT5ParserHelper::doEndCurve</b>	<b>97</b>
<b>AIRINV::FRAT5ParserHelper::storeCurveKey</b>	<b>251</b>
<b>AIRINV::FRAT5ParserHelper::storeDTD</b>	<b>265</b>
<b>AIRINV::FRAT5ParserHelper::storeFRAT5Value</b>	<b>291</b>
<b>ParsingFileFailedException</b>	<b>199</b>
<b>AIRINV::FFDisutilityFileParsingFailedException</b>	<b>117</b>
<b>AIRINV::FRAT5FileParsingFailedException</b>	<b>155</b>
<b>AIRINV::InventoryFileParsingFailedException</b>	<b>165</b>
<b>AIRINV::ScheduleFileParsingFailedException</b>	<b>204</b>
<b>AIRINV::MissingPartnerFlightDateWithinScheduleFile</b>	<b>190</b>
<b>std::priority_queue&lt; T &gt;</b>	
<b>std::queue&lt; T &gt;</b>	
<b>AIRINV::Reply</b>	<b>199</b>
<b>AIRINV::Request</b>	<b>200</b>
<b>AIRINV::RequestParser</b>	<b>203</b>

<b>RootException</b>	<b>204</b>
<b>AIRINV::BookingException</b>	<b>75</b>
<b>AIRINV::SegmentCabinHelper</b>	<b>206</b>
<b>AIRINV::SegmentDateHelper</b>	<b>210</b>
<b>AIRINV::SegmentSnapshotTableHelper</b>	<b>211</b>
<b>AIRINV::ServiceAbstract</b>	<b>214</b>
<b>ServiceAbstract</b>	<b>215</b>
<b>AIRINV::AIRINV_Master_ServiceContext</b>	<b>56</b>
<b>AIRINV::AIRINV_ServiceContext</b>	<b>65</b>
std::set< K >	
std::stack< T >	
<b>StructAbstract</b>	<b>375</b>
<b>AIRINV::BookingClassStruct</b>	<b>71</b>
<b>AIRINV::BucketStruct</b>	<b>76</b>
<b>AIRINV::DCPEventStruct</b>	<b>82</b>
<b>AIRINV::FareFamilyStruct</b>	<b>114</b>
<b>AIRINV::FFDisutilityStruct</b>	<b>122</b>
<b>AIRINV::FlightDateStruct</b>	<b>127</b>
<b>AIRINV::FlightPeriodStruct</b>	<b>141</b>
<b>AIRINV::FlightRequestStatus</b>	<b>148</b>
<b>AIRINV::FlightTypeCode</b>	<b>150</b>
<b>AIRINV::FlightVisibilityCode</b>	<b>152</b>
<b>AIRINV::FRAT5Struct</b>	<b>160</b>
<b>AIRINV::LegCabinStruct</b>	<b>183</b>
<b>AIRINV::LegStruct</b>	<b>186</b>
<b>AIRINV::SegmentCabinStruct</b>	<b>208</b>
<b>AIRINV::SegmentStruct</b>	<b>212</b>
std::valarray< T >	
std::vector< T >	

## 5 AirInv Class Index

### 5.1 AirInv Class List

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

<a href="#"><b>AIRINV::AIRINV_Master_Service</b> (Interface for the <b>AIRINV</b> Services )</a>	<b>47</b>
<a href="#"><b>AIRINV::AIRINV_Master_ServiceContext</b></a>	<b>56</b>
<a href="#"><b>AIRINV::AIRINV_Service</b> (Interface for the <b>AIRINV</b> Services )</a>	<b>57</b>
<a href="#"><b>AIRINV::AIRINV_ServiceContext</b> (Class holding the context of the AirInv services )</a>	<b>65</b>
<a href="#"><b>AIRINV::AirInvServer</b></a>	<b>66</b>
<a href="#"><b>AIRINV::BomAbstract</b></a>	<b>67</b>
<a href="#"><b>stdair::BomPropertyTree</b></a>	<b>69</b>
<a href="#"><b>AIRINV::BomRootHelper</b></a>	<b>70</b>
<a href="#"><b>AIRINV::BookingClassHelper</b></a>	<b>71</b>
<a href="#"><b>AIRINV::BookingClassStruct</b></a>	<b>71</b>
<a href="#"><b>AIRINV::BookingException</b></a>	<b>75</b>
<a href="#"><b>enable_shared_from_this</b></a>	<b>75</b>
<a href="#"><b>noncopyable</b></a>	<b>75</b>
<a href="#"><b>grammar</b></a>	<b>76</b>
<a href="#"><b>grammar</b></a>	<b>76</b>
<a href="#"><b>AIRINV::BucketStruct</b> (Utility Structure for the parsing of Bucket structures )</a>	<b>76</b>
<a href="#"><b>CmdAbstract</b></a>	<b>78</b>
<a href="#"><b>AIRINV::Connection</b></a>	<b>79</b>
<a href="#"><b>TestFixture</b></a>	<b>80</b>
<a href="#"><b>AIRINV::DCPEventGenerator</b></a>	<b>81</b>
<a href="#"><b>AIRINV::DCPEventStruct</b></a>	<b>82</b>
<a href="#"><b>AIRINV::DCPParser</b></a>	<b>89</b>
<a href="#"><b>AIRINV::DCPRuleFileParser</b></a>	<b>90</b>
<a href="#"><b>AIRINV::DCPParserHelper::DCPRuleParser</b></a>	<b>91</b>
<a href="#"><b>AIRINV::DefaultMap</b></a>	<b>95</b>
<a href="#"><b>AIRINV::FFDisutilityParserHelper::doEndCurve</b></a>	<b>96</b>

<b>AIRINV::FRAT5ParserHelper::doEndCurve</b>	<b>97</b>
<b>AIRINV::DCPParserHelper::doEndDCP</b>	<b>98</b>
<b>AIRINV::ScheduleParserHelper::doEndFlight</b>	<b>100</b>
<b>AIRINV::InventoryParserHelper::doEndFlightDate</b>	<b>102</b>
<b>AIRINV::FacAirinvMasterServiceContext (Factory for Bucket )</b>	<b>104</b>
<b>AIRINV::FacAirinvServiceContext</b>	<b>105</b>
<b>AIRINV::FacBomAbstract</b>	<b>107</b>
<b>FacServiceAbstract</b>	<b>109</b>
<b>AIRINV::FacServiceAbstract</b>	<b>109</b>
<b>AIRINV::FacSupervisor</b>	<b>111</b>
<b>AIRINV::FareFamilyStruct (Utility Structure for the parsing of fare family details )</b>	<b>114</b>
<b>AIRINV::FFDisutilityFileParser</b>	<b>116</b>
<b>AIRINV::FFDisutilityFileParsingFailedException</b>	<b>117</b>
<b>AIRINV::FFDisutilityInputFileNotFoundException</b>	<b>117</b>
<b>AIRINV::FFDisutilityParser (Class wrapping the parser entry point )</b>	<b>118</b>
<b>AIRINV::FFDisutilityParserHelper::FFDisutilityParser</b>	<b>119</b>
<b>AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition&lt; ScannerT &gt;</b>	<b>120</b>
<b>AIRINV::FFDisutilityStruct</b>	<b>122</b>
<b>FileNotFoundExpection</b>	<b>124</b>
<b>AIRINV::FlightDateDuplicationException</b>	<b>124</b>
<b>AIRINV::FlightDateHelper</b>	<b>125</b>
<b>AIRINV::FlightDateNotFoundExpection</b>	<b>126</b>
<b>AIRINV::FlightDateStruct</b>	<b>127</b>
<b>AIRINV::FlightPeriodFileParser</b>	<b>134</b>
<b>AIRINV::ScheduleParserHelper::FlightPeriodParser</b>	<b>135</b>
<b>AIRINV::ScheduleParserHelper::FlightPeriodParser::definition&lt; ScannerT &gt;</b>	<b>136</b>
<b>AIRINV::FlightPeriodStruct</b>	<b>141</b>
<b>AIRINV::FlightRequestStatus</b>	<b>148</b>
<b>AIRINV::FlightTypeCode</b>	<b>150</b>

<a href="#">AIRINV::FlightVisibilityCode</a>	<a href="#">152</a>
<a href="#">AIRINV::FRAT5FileParser</a>	<a href="#">154</a>
<a href="#">AIRINV::FRAT5FileParsingFailedException</a>	<a href="#">155</a>
<a href="#">AIRINV::FRAT5InputFileNotFoundException</a>	<a href="#">156</a>
<a href="#">AIRINV::FRAT5Parser (Class wrapping the parser entry point )</a>	<a href="#">156</a>
<a href="#">AIRINV::FRAT5ParserHelper::FRAT5Parser</a>	<a href="#">157</a>
<a href="#">AIRINV::FRAT5ParserHelper::FRAT5Parser::definition&lt; ScannerT &gt;</a>	<a href="#">158</a>
<a href="#">AIRINV::FRAT5Struct</a>	<a href="#">160</a>
<a href="#">AIRINV::header</a>	<a href="#">162</a>
<a href="#">InputFilePath</a>	<a href="#">163</a>
<a href="#">AIRINV::InventoryBuilder (Class handling the generation / instantiation of the Inventory BOM )</a>	<a href="#">163</a>
<a href="#">AIRINV::InventoryFileParser</a>	<a href="#">164</a>
<a href="#">AIRINV::InventoryFileParsingFailedException</a>	<a href="#">165</a>
<a href="#">AIRINV::InventoryFilePath</a>	<a href="#">165</a>
<a href="#">AIRINV::InventoryGenerator (Class handling the generation / instantiation of the Inventory BOM )</a>	<a href="#">166</a>
<a href="#">AIRINV::InventoryHelper</a>	<a href="#">167</a>
<a href="#">AIRINV::InventoryInputFileNotFoundException</a>	<a href="#">169</a>
<a href="#">AIRINV::InventoryManager</a>	<a href="#">169</a>
<a href="#">AIRINV::InventoryNotFoundException</a>	<a href="#">172</a>
<a href="#">AIRINV::InventoryParserHelper::InventoryParser</a>	<a href="#">173</a>
<a href="#">AIRINV::InventoryParser (Class wrapping the parser entry point )</a>	<a href="#">174</a>
<a href="#">AIRINV::InventoryParserHelper::InventoryParser::definition&lt; ScannerT &gt;</a>	<a href="#">175</a>
<a href="#">InventoryTestSuite</a>	<a href="#">182</a>
<a href="#">AIRINV::LegCabinHelper</a>	<a href="#">183</a>
<a href="#">AIRINV::LegCabinStruct</a>	<a href="#">183</a>
<a href="#">AIRINV::LegStruct</a>	<a href="#">186</a>
<a href="#">AIRINV::MissingPartnerFlightDateWithinScheduleFile</a>	<a href="#">190</a>
<a href="#">ObjectCreationDuplicationException</a>	<a href="#">190</a>



<b>ObjectNotFoundException</b>	<b>191</b>
<b>ParserException</b>	<b>191</b>
<b>AIRINV::InventoryParserHelper::ParserSemanticAction</b>	<b>191</b>
<b>AIRINV::ScheduleParserHelper::ParserSemanticAction</b>	<b>193</b>
<b>AIRINV::DCPParserHelper::ParserSemanticAction</b>	<b>195</b>
<b>AIRINV::FFDisutilityParserHelper::ParserSemanticAction</b>	<b>197</b>
<b>AIRINV::FRAT5ParserHelper::ParserSemanticAction</b>	<b>198</b>
<b>ParsingFileFailedException</b>	<b>199</b>
<b>AIRINV::Reply</b>	<b>199</b>
<b>AIRINV::Request</b>	<b>200</b>
<b>AIRINV::RequestHandler</b> (The common handler for all incoming requests )	<b>202</b>
<b>AIRINV::RequestParser</b> (Parser for incoming requests )	<b>203</b>
<b>RootException</b>	<b>204</b>
<b>AIRINV::ScheduleFileParsingFailedException</b>	<b>204</b>
<b>AIRINV::ScheduleInputFileNotFoundException</b>	<b>205</b>
<b>AIRINV::ScheduleParser</b> (Class wrapping the parser entry point )	<b>205</b>
<b>AIRINV::SegmentCabinHelper</b> (Class representing the actual business functions for an air-line segment-cabin )	<b>206</b>
<b>AIRINV::SegmentCabinStruct</b> (Utility Structure for the parsing of SegmentCabin details )	<b>208</b>
<b>AIRINV::SegmentDateHelper</b>	<b>210</b>
<b>AIRINV::SegmentDateNotFoundException</b>	<b>211</b>
<b>AIRINV::SegmentSnapshotTableHelper</b>	<b>211</b>
<b>AIRINV::SegmentStruct</b>	<b>212</b>
<b>AIRINV::ServiceAbstract</b>	<b>214</b>
<b>ServiceAbstract</b>	<b>215</b>
<b>AIRINV::InventoryParserHelper::storeACP</b>	<b>216</b>
<b>AIRINV::DCPParserHelper::storeAdvancePurchase</b>	<b>217</b>
<b>AIRINV::InventoryParserHelper::storeAirlineCode</b>	<b>219</b>
<b>AIRINV::DCPParserHelper::storeAirlineCode</b>	<b>221</b>
<b>AIRINV::ScheduleParserHelper::storeAirlineCode</b>	<b>222</b>

AIRINV::InventoryParserHelper::storeAU	224
AIRINV::InventoryParserHelper::storeBoardingDate	225
AIRINV::ScheduleParserHelper::storeBoardingTime	227
AIRINV::InventoryParserHelper::storeBoardingTime	229
AIRINV::InventoryParserHelper::storeBookingCounter	231
AIRINV::InventoryParserHelper::storeBucketAvailability	232
AIRINV::DCPParserHelper::storeCabinCode	234
AIRINV::ScheduleParserHelper::storeCapacity	236
AIRINV::DCPParserHelper::storeChangeFees	237
AIRINV::DCPParserHelper::storeChannel	238
AIRINV::DCPParserHelper::storeClass	240
AIRINV::InventoryParserHelper::storeClassAvailability	241
AIRINV::InventoryParserHelper::storeClassCode	243
AIRINV::ScheduleParserHelper::storeClasses	245
AIRINV::InventoryParserHelper::storeClassETB	246
AIRINV::InventoryParserHelper::storeCumulatedProtection	248
AIRINV::FFDisutilityParserHelper::storeCurveKey	250
AIRINV::FRAT5ParserHelper::storeCurveKey	251
AIRINV::DCPParserHelper::storeDateRangeEnd	253
AIRINV::ScheduleParserHelper::storeDateRangeEnd	254
AIRINV::ScheduleParserHelper::storeDateRangeStart	256
AIRINV::DCPParserHelper::storeDateRangeStart	257
AIRINV::DCPParserHelper::storeDCP	258
AIRINV::DCPParserHelper::storeDCPId	260
AIRINV::DCPParserHelper::storeDestination	261
AIRINV::ScheduleParserHelper::storeDow	263
AIRINV::FFDisutilityParserHelper::storeDTD	264
AIRINV::FRAT5ParserHelper::storeDTD	265
AIRINV::ScheduleParserHelper::storeElapsedTime	266

AIRINV::DCPParserHelper::storeEndRangeTime	268
AIRINV::InventoryParserHelper::storeETB	269
AIRINV::ScheduleParserHelper::storeFamilyCode	271
AIRINV::InventoryParserHelper::storeFamilyCode	273
AIRINV::ScheduleParserHelper::storeFClasses	275
AIRINV::InventoryParserHelper::storeFClasses	276
AIRINV::ScheduleParserHelper::storeFFDisutilityCurveKey	278
AIRINV::FFDisutilityParserHelper::storeFFDisutilityValue	280
AIRINV::InventoryParserHelper::storeFlightDate	281
AIRINV::InventoryParserHelper::storeFlightNumber	283
AIRINV::ScheduleParserHelper::storeFlightNumber	284
AIRINV::InventoryParserHelper::storeFlightTypeCode	286
AIRINV::InventoryParserHelper::storeFlightVisibilityCode	288
AIRINV::ScheduleParserHelper::storeFRAT5CurveKey	290
AIRINV::FRAT5ParserHelper::storeFRAT5Value	291
AIRINV::InventoryParserHelper::storeGAV	292
AIRINV::InventoryParserHelper::storeLegBoardingPoint	294
AIRINV::ScheduleParserHelper::storeLegBoardingPoint	296
AIRINV::ScheduleParserHelper::storeLegCabinCode	298
AIRINV::InventoryParserHelper::storeLegCabinCode	299
AIRINV::ScheduleParserHelper::storeLegOffPoint	301
AIRINV::InventoryParserHelper::storeLegOffPoint	302
AIRINV::DCPParserHelper::storeMinimumStay	304
AIRINV::InventoryParserHelper::storeNAV	305
AIRINV::InventoryParserHelper::storeNbOfBkgs	307
AIRINV::InventoryParserHelper::storeNbOfGroupBkgs	309
AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs	311
AIRINV::InventoryParserHelper::storeNbOfStaffBkgs	313
AIRINV::InventoryParserHelper::storeNbOfWLBkgs	315

AIRINV::InventoryParserHelper::storeNego	317
AIRINV::DCPParserHelper::storeNonRefundable	318
AIRINV::InventoryParserHelper::storeNoShow	320
AIRINV::InventoryParserHelper::storeOffDate	322
AIRINV::ScheduleParserHelper::storeOffTime	323
AIRINV::InventoryParserHelper::storeOffTime	325
AIRINV::ScheduleParserHelper::storeOperatingAirlineCode	327
AIRINV::InventoryParserHelper::storeOperatingAirlineCode	328
AIRINV::ScheduleParserHelper::storeOperatingFlightNumber	330
AIRINV::InventoryParserHelper::storeOperatingFlightNumber	332
AIRINV::DCPParserHelper::storeOrigin	334
AIRINV::InventoryParserHelper::storeOverbooking	335
AIRINV::InventoryParserHelper::storeParentClassCode	337
AIRINV::InventoryParserHelper::storeParentSubclassCode	339
AIRINV::DCPParserHelper::storePOS	341
AIRINV::InventoryParserHelper::storeProtection	342
AIRINV::InventoryParserHelper::storeRevenueAvailability	344
AIRINV::InventoryParserHelper::storeSaleableCapacity	346
AIRINV::DCPParserHelper::storeSaturdayStay	347
AIRINV::InventoryParserHelper::storeSeatIndex	349
AIRINV::InventoryParserHelper::storeSegmentAvailability	351
AIRINV::ScheduleParserHelper::storeSegmentBoardingPoint	352
AIRINV::InventoryParserHelper::storeSegmentBoardingPoint	354
AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter	356
AIRINV::InventoryParserHelper::storeSegmentCabinCode	358
AIRINV::ScheduleParserHelper::storeSegmentCabinCode	360
AIRINV::InventoryParserHelper::storeSegmentOffPoint	361
AIRINV::ScheduleParserHelper::storeSegmentOffPoint	363
AIRINV::ScheduleParserHelper::storeSegmentSpecificity	365

<a href="#">AIRINV::InventoryParserHelper::storeSnapshotDate</a>	366
<a href="#">AIRINV::DCPPParserHelper::storeStartRangeTime</a>	368
<a href="#">AIRINV::InventoryParserHelper::storeSubclassCode</a>	369
<a href="#">AIRINV::InventoryParserHelper::storeUPR</a>	371
<a href="#">AIRINV::InventoryParserHelper::storeYieldUpperRange</a>	373
<a href="#">StructAbstract</a>	375

## 6 AirInv File Index

### 6.1 AirInv File List

Here is a list of all files with brief descriptions:

<a href="#">airinv/AIRINV_Master_Service.hpp</a>	377
<a href="#">airinv/AIRINV_Service.hpp</a>	377
<a href="#">airinv/AIRINV_Types.hpp</a>	378
<a href="#">airinv/FlightRequestStatus.hpp</a>	418
<a href="#">airinv/basic/BasConst.cpp</a>	379
<a href="#">airinv/basic/BasConst_AIRINV_Service.hpp</a>	379
<a href="#">airinv/basic/BasConst_Curves.hpp</a>	379
<a href="#">airinv/basic/BasConst_General.hpp</a>	380
<a href="#">airinv/basic/BasParserTypes.hpp</a>	380
<a href="#">airinv/basic/FlightRequestStatus.cpp</a>	381
<a href="#">airinv/basic/FlightTypeCode.cpp</a>	381
<a href="#">airinv/basic/FlightTypeCode.hpp</a>	381
<a href="#">airinv/basic/FlightVisibilityCode.cpp</a>	381
<a href="#">airinv/basic/FlightVisibilityCode.hpp</a>	382
<a href="#">airinv/batches/airinv_parseInventory.cpp</a>	382
<a href="#">airinv/batches/parseInventory.cpp</a>	382
<a href="#">airinv/bom/AirportList.hpp</a>	382
<a href="#">airinv/bom/BomAbstract.cpp</a>	382
<a href="#">airinv/bom/BomAbstract.hpp</a>	383

<a href="#">airinv/bom/BomRootHelper.cpp</a>	383
<a href="#">airinv/bom/BomRootHelper.hpp</a>	384
<a href="#">airinv/bom/BookingClassHelper.cpp</a>	384
<a href="#">airinv/bom/BookingClassHelper.hpp</a>	384
<a href="#">airinv/bom/BookingClassStruct.cpp</a>	384
<a href="#">airinv/bom/BookingClassStruct.hpp</a>	385
<a href="#">airinv/bom/BucketStruct.cpp</a>	385
<a href="#">airinv/bom/BucketStruct.hpp</a>	385
<a href="#">airinv/bom/DCPEventStruct.cpp</a>	386
<a href="#">airinv/bom/DCPEventStruct.hpp</a>	386
<a href="#">airinv/bom/FareFamilyStruct.cpp</a>	387
<a href="#">airinv/bom/FareFamilyStruct.hpp</a>	387
<a href="#">airinv/bom/FFDisutilityStruct.cpp</a>	387
<a href="#">airinv/bom/FFDisutilityStruct.hpp</a>	388
<a href="#">airinv/bom/FlightDateHelper.cpp</a>	388
<a href="#">airinv/bom/FlightDateHelper.hpp</a>	388
<a href="#">airinv/bom/FlightDateStruct.cpp</a>	389
<a href="#">airinv/bom/FlightDateStruct.hpp</a>	389
<a href="#">airinv/bom/FlightPeriodStruct.cpp</a>	390
<a href="#">airinv/bom/FlightPeriodStruct.hpp</a>	390
<a href="#">airinv/bom/FRAT5Struct.cpp</a>	390
<a href="#">airinv/bom/FRAT5Struct.hpp</a>	391
<a href="#">airinv/bom/InventoryHelper.cpp</a>	391
<a href="#">airinv/bom/InventoryHelper.hpp</a>	392
<a href="#">airinv/bom/LegCabinHelper.cpp</a>	392
<a href="#">airinv/bom/LegCabinHelper.hpp</a>	392
<a href="#">airinv/bom/LegCabinStruct.cpp</a>	392
<a href="#">airinv/bom/LegCabinStruct.hpp</a>	393
<a href="#">airinv/bom/LegStruct.cpp</a>	393

<a href="#">airinv/bom/LegStruct.hpp</a>	393
<a href="#">airinv/bom/SegmentCabinHelper.cpp</a>	394
<a href="#">airinv/bom/SegmentCabinHelper.hpp</a>	394
<a href="#">airinv/bom/SegmentCabinStruct.cpp</a>	395
<a href="#">airinv/bom/SegmentCabinStruct.hpp</a>	395
<a href="#">airinv/bom/SegmentDateHelper.cpp</a>	396
<a href="#">airinv/bom/SegmentDateHelper.hpp</a>	396
<a href="#">airinv/bom/SegmentSnapshotTableHelper.cpp</a>	396
<a href="#">airinv/bom/SegmentSnapshotTableHelper.hpp</a>	397
<a href="#">airinv/bom/SegmentStruct.cpp</a>	397
<a href="#">airinv/bom/SegmentStruct.hpp</a>	397
<a href="#">airinv/command/FFDisutilityParser.cpp</a>	398
<a href="#">airinv/command/FFDisutilityParser.hpp</a>	398
<a href="#">airinv/command/FFDisutilityParserHelper.cpp</a>	398
<a href="#">airinv/command/FFDisutilityParserHelper.hpp</a>	399
<a href="#">airinv/command/FRAT5Parser.cpp</a>	399
<a href="#">airinv/command/FRAT5Parser.hpp</a>	400
<a href="#">airinv/command/FRAT5ParserHelper.cpp</a>	400
<a href="#">airinv/command/FRAT5ParserHelper.hpp</a>	401
<a href="#">airinv/command/InventoryBuilder.cpp</a>	401
<a href="#">airinv/command/InventoryBuilder.hpp</a>	402
<a href="#">airinv/command/InventoryGenerator.cpp</a>	402
<a href="#">airinv/command/InventoryGenerator.hpp</a>	403
<a href="#">airinv/command/InventoryManager.cpp</a>	404
<a href="#">airinv/command/InventoryManager.hpp</a>	405
<a href="#">airinv/command/InventoryParser.cpp</a>	405
<a href="#">airinv/command/InventoryParser.hpp</a>	406
<a href="#">airinv/command/InventoryParserHelper.cpp</a>	406
<a href="#">airinv/command/InventoryParserHelper.hpp</a>	407

<a href="#">airinv/command/ScheduleParser.cpp</a>	408
<a href="#">airinv/command/ScheduleParser.hpp</a>	409
<a href="#">airinv/command/ScheduleParserHelper.cpp</a>	409
<a href="#">airinv/command/ScheduleParserHelper.hpp</a>	410
<a href="#">airinv/command/vault/DCPEventGenerator.cpp</a>	411
<a href="#">airinv/command/vault/DCPEventGenerator.hpp</a>	411
<a href="#">airinv/command/vault/DCPParser.cpp</a>	412
<a href="#">airinv/command/vault/DCPParser.hpp</a>	412
<a href="#">airinv/command/vault/DCPParserHelper.cpp</a>	412
<a href="#">airinv/command/vault/DCPParserHelper.hpp</a>	413
<a href="#">airinv/config/airinv-paths.hpp.in</a>	414
<a href="#">airinv/factory/FacAirinvMasterServiceContext.cpp</a>	416
<a href="#">airinv/factory/FacAirinvMasterServiceContext.hpp</a>	416
<a href="#">airinv/factory/FacAirinvServiceContext.cpp</a>	416
<a href="#">airinv/factory/FacAirinvServiceContext.hpp</a>	416
<a href="#">airinv/factory/FacBomAbstract.cpp</a>	417
<a href="#">airinv/factory/FacBomAbstract.hpp</a>	417
<a href="#">airinv/factory/FacServiceAbstract.cpp</a>	417
<a href="#">airinv/factory/FacServiceAbstract.hpp</a>	417
<a href="#">airinv/factory/FacSupervisor.cpp</a>	418
<a href="#">airinv/factory/FacSupervisor.hpp</a>	418
<a href="#">airinv/server/AirInvClient.cpp</a>	419
<a href="#">airinv/server/AirInvClient_ASIO.cpp</a>	419
<a href="#">airinv/server/AirInvServer.cpp</a>	419
<a href="#">airinv/server/AirInvServer.hpp</a>	419
<a href="#">airinv/server/AirInvServer_ASIO.cpp</a>	420
<a href="#">airinv/server/BomPropertyTree.cpp</a>	420
<a href="#">airinv/server/BomPropertyTree.hpp</a>	420
<a href="#">airinv/server/Connection.cpp</a>	421



<a href="#">airinv/server/Connection.hpp</a>	421
<a href="#">airinv/server/header.hpp</a>	422
<a href="#">airinv/server/posix_main.cpp</a>	422
<a href="#">airinv/server/Reply.cpp</a>	422
<a href="#">airinv/server/Reply.hpp</a>	423
<a href="#">airinv/server/Request.cpp</a>	423
<a href="#">airinv/server/Request.hpp</a>	423
<a href="#">airinv/server/RequestHandler.cpp</a>	424
<a href="#">airinv/server/RequestHandler.hpp</a>	424
<a href="#">airinv/server/RequestParser.cpp</a>	424
<a href="#">airinv/server/RequestParser.hpp</a>	425
<a href="#">airinv/server/win_main.cpp</a>	425
<a href="#">airinv/service/AIRINV_Master_Service.cpp</a>	425
<a href="#">airinv/service/AIRINV_Master_ServiceContext.cpp</a>	426
<a href="#">airinv/service/AIRINV_Master_ServiceContext.hpp</a>	426
<a href="#">airinv/service/AIRINV_Service.cpp</a>	426
<a href="#">airinv/service/AIRINV_ServiceContext.cpp</a>	428
<a href="#">airinv/service/AIRINV_ServiceContext.hpp</a>	428
<a href="#">airinv/service/ServiceAbstract.cpp</a>	428
<a href="#">airinv/service/ServiceAbstract.hpp</a>	428
<a href="#">airinv/ui/cmdline/airinv.cpp</a>	430
<a href="#">test/airinv/InventoryTestSuite.cpp</a>	430
<a href="#">test/airinv/InventoryTestSuite.hpp</a>	430

## 7 AirInv Page Index

### 7.1 AirInv Related Pages

Here is a list of all related documentation pages:

<a href="#">People</a>	431
<a href="#">Coding Rules</a>	431

<b>Copyright and License</b>	<b><a href="#">432</a></b>
<b>Documentation Rules</b>	<b><a href="#">439</a></b>
<b>Main features</b>	<b><a href="#">440</a></b>
<b>Make a Difference</b>	<b><a href="#">441</a></b>
<b>Make a new release</b>	<b><a href="#">442</a></b>
<b>Installation</b>	<b><a href="#">444</a></b>
<b>Linking with Airinv</b>	<b><a href="#">455</a></b>
<b>Test Rules</b>	<b><a href="#">457</a></b>
<b>Users Guide</b>	<b><a href="#">458</a></b>
<b>Supported Systems</b>	<b><a href="#">509</a></b>
<b>AirInv Supported Systems (Previous Releases)</b>	<b><a href="#">510</a></b>
<b>Tutorials</b>	<b><a href="#">514</a></b>
<b>Command-Line Test to Demonstrate How To Test the AirInv Project</b>	<b><a href="#">517</a></b>

## 8 AirInv Directory Documentation

### 8.1 test/airinv/ Directory Reference

#### Files

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

### 8.2 airinv/ Directory Reference

#### Directories

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

**Files**

- file [AIRINV\\_Master\\_Service.hpp](#)
- file [AIRINV\\_Service.hpp](#)
- file [AIRINV\\_Types.hpp](#)
- file [FlightRequestStatus.hpp](#)

**8.3 airinv/basic/ Directory Reference****Files**

- file [BasConst.cpp](#)
- file [BasConst\\_AIRINV\\_Service.hpp](#)
- file [BasConst\\_Curves.hpp](#)
- file [BasConst\\_General.hpp](#)
- file [BasParserTypes.hpp](#)
- file [FlightRequestStatus.cpp](#)
- file [FlightTypeCode.cpp](#)
- file [FlightTypeCode.hpp](#)
- file [FlightVisibilityCode.cpp](#)
- file [FlightVisibilityCode.hpp](#)

**8.4 airinv/batches/ Directory Reference****Files**

- file [airinv\\_parseInventory.cpp](#)
- file [parseInventory.cpp](#)

**8.5 airinv/bom/ Directory Reference****Files**

- file [AirportList.hpp](#)
- file [BomAbstract.cpp](#)
- file [BomAbstract.hpp](#)
- file [BomRootHelper.cpp](#)
- file [BomRootHelper.hpp](#)
- file [BookingClassHelper.cpp](#)
- file [BookingClassHelper.hpp](#)
- file [BookingClassStruct.cpp](#)
- file [BookingClassStruct.hpp](#)
- file [BucketStruct.cpp](#)
- file [BucketStruct.hpp](#)
- file [DCPEventStruct.cpp](#)
- file [DCPEventStruct.hpp](#)
- file [FareFamilyStruct.cpp](#)
- file [FareFamilyStruct.hpp](#)
- file [FFDisutilityStruct.cpp](#)

- file [FFDisutilityStruct.hpp](#)
- file [FlightDateHelper.cpp](#)
- file [FlightDateHelper.hpp](#)
- file [FlightDateStruct.cpp](#)
- file [FlightDateStruct.hpp](#)
- file [FlightPeriodStruct.cpp](#)
- file [FlightPeriodStruct.hpp](#)
- file [FRAT5Struct.cpp](#)
- file [FRAT5Struct.hpp](#)
- file [InventoryHelper.cpp](#)
- file [InventoryHelper.hpp](#)
- file [LegCabinHelper.cpp](#)
- file [LegCabinHelper.hpp](#)
- file [LegCabinStruct.cpp](#)
- file [LegCabinStruct.hpp](#)
- file [LegStruct.cpp](#)
- file [LegStruct.hpp](#)
- file [SegmentCabinHelper.cpp](#)
- file [SegmentCabinHelper.hpp](#)
- file [SegmentCabinStruct.cpp](#)
- file [SegmentCabinStruct.hpp](#)
- file [SegmentDateHelper.cpp](#)
- file [SegmentDateHelper.hpp](#)
- file [SegmentSnapshotTableHelper.cpp](#)
- file [SegmentSnapshotTableHelper.hpp](#)
- file [SegmentStruct.cpp](#)
- file [SegmentStruct.hpp](#)

## 8.6 airinv/ui/cmdline/ Directory Reference

### Files

- file [airinv.cpp](#)

## 8.7 airinv/command/ Directory Reference

### Directories

- directory [vault](#)

### Files

- file [FFDisutilityParser.cpp](#)
- file [FFDisutilityParser.hpp](#)
- file [FFDisutilityParserHelper.cpp](#)
- file [FFDisutilityParserHelper.hpp](#)
- file [FRAT5Parser.cpp](#)
- file [FRAT5Parser.hpp](#)
- file [FRAT5ParserHelper.cpp](#)

- file [FRAT5ParserHelper.hpp](#)
- file [InventoryBuilder.cpp](#)
- file [InventoryBuilder.hpp](#)
- file [InventoryGenerator.cpp](#)
- file [InventoryGenerator.hpp](#)
- file [InventoryManager.cpp](#)
- file [InventoryManager.hpp](#)
- file [InventoryParser.cpp](#)
- file [InventoryParser.hpp](#)
- file [InventoryParserHelper.cpp](#)
- file [InventoryParserHelper.hpp](#)
- file [ScheduleParser.cpp](#)
- file [ScheduleParser.hpp](#)
- file [ScheduleParserHelper.cpp](#)
- file [ScheduleParserHelper.hpp](#)

## 8.8 airinv/config/ Directory Reference

### Files

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

## 8.9 airinv/factory/ Directory Reference

### Files

- file [FacAirinvMasterServiceContext.cpp](#)
- file [FacAirinvMasterServiceContext.hpp](#)
- file [FacAirinvServiceContext.cpp](#)
- file [FacAirinvServiceContext.hpp](#)
- file [FacBomAbstract.cpp](#)
- file [FacBomAbstract.hpp](#)
- file [FacServiceAbstract.cpp](#)
- file [FacServiceAbstract.hpp](#)
- file [FacSupervisor.cpp](#)
- file [FacSupervisor.hpp](#)

## 8.10 airinv/server/ Directory Reference

### Files

- file [AirInvClient.cpp](#)
- file [AirInvClient\\_ASIO.cpp](#)
- file [AirInvServer.cpp](#)
- file [AirInvServer.hpp](#)
- file [AirInvServer\\_ASIO.cpp](#)
- file [BomPropertyTree.cpp](#)
- file [BomPropertyTree.hpp](#)
- file [Connection.cpp](#)

- file [Connection.hpp](#)
- file [header.hpp](#)
- file [posix\\_main.cpp](#)
- file [Reply.cpp](#)
- file [Reply.hpp](#)
- file [Request.cpp](#)
- file [Request.hpp](#)
- file [RequestHandler.cpp](#)
- file [RequestHandler.hpp](#)
- file [RequestParser.cpp](#)
- file [RequestParser.hpp](#)
- file [win\\_main.cpp](#)

## 8.11 airinv/service/ Directory Reference

### Files

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

## 8.12 test/ Directory Reference

### Directories

- directory [airinv](#)

## 8.13 airinv/ui/ Directory Reference

### Directories

- directory [cmdline](#)

## 8.14 airinv/command/vault/ Directory Reference

### Files

- file [DCPEventGenerator.cpp](#)
- file [DCPEventGenerator.hpp](#)
- file [DCPParser.cpp](#)
- file [DCPParser.hpp](#)
- file [DCPParserHelper.cpp](#)
- file [DCPParserHelper.hpp](#)

## 9 AirInv Namespace Documentation

### 9.1 AIRINV Namespace Reference

#### Classes

- class [AIRINV\\_Master\\_Service](#)  
*Interface for the [AIRINV](#) Services.*
- class [AIRINV\\_Service](#)  
*Interface for the [AIRINV](#) Services.*
- class [InventoryFileParsingFailedException](#)
- class [ScheduleFileParsingFailedException](#)
- class [MissingPartnerFlightDateWithinScheduleFile](#)
- class [FRAT5FileParsingFailedException](#)
- class [FFDisutilityFileParsingFailedException](#)
- class [SegmentDateNotFoundException](#)
- class [InventoryInputFileNotFoundException](#)
- class [ScheduleInputFileNotFoundException](#)
- class [FRAT5InputFileNotFoundException](#)
- class [FFDisutilityInputFileNotFoundException](#)
- class [FlightDateDuplicationException](#)
- class [BookingException](#)
- class [InventoryNotFoundException](#)
- class [FlightDateNotFoundException](#)
- class [InventoryFilePath](#)
- struct [DefaultMap](#)
- struct [FlightTypeCode](#)
- struct [FlightVisibilityCode](#)
- class [BomAbstract](#)
- class [BomRootHelper](#)
- class [BookingClassHelper](#)
- struct [BookingClassStruct](#)
- struct [BucketStruct](#)  
*Utility Structure for the parsing of Bucket structures.*
- struct [DCPEventStruct](#)
- struct [FareFamilyStruct](#)  
*Utility Structure for the parsing of fare family details.*
- struct [FFDisutilityStruct](#)
- class [FlightDateHelper](#)
- struct [FlightDateStruct](#)
- struct [FlightPeriodStruct](#)
- struct [FRAT5Struct](#)
- class [InventoryHelper](#)
- class [LegCabinHelper](#)
- struct [LegCabinStruct](#)
- struct [LegStruct](#)

- class [SegmentCabinHelper](#)  
*Class representing the actual business functions for an airline segment-cabin.*
- struct [SegmentCabinStruct](#)  
*Utility Structure for the parsing of SegmentCabin details.*
- class [SegmentDateHelper](#)
- class [SegmentSnapshotTableHelper](#)
- struct [SegmentStruct](#)
- class [FFDisutilityParser](#)  
*Class wrapping the parser entry point.*
- class [FFDisutilityFileParser](#)
- class [FRAT5Parser](#)  
*Class wrapping the parser entry point.*
- class [FRAT5FileParser](#)
- class [InventoryBuilder](#)  
*Class handling the generation / instantiation of the Inventory BOM.*
- class [InventoryGenerator](#)  
*Class handling the generation / instantiation of the Inventory BOM.*
- class [InventoryManager](#)
- class [InventoryParser](#)  
*Class wrapping the parser entry point.*
- class [InventoryFileParser](#)
- class [ScheduleParser](#)  
*Class wrapping the parser entry point.*
- class [FlightPeriodFileParser](#)
- class [DCPEventGenerator](#)
- class [DCPParser](#)
- class [DCPRuleFileParser](#)
- class [FacAirinvMasterServiceContext](#)  
*Factory for Bucket.*
- class [FacAirinvServiceContext](#)
- class [FacBomAbstract](#)
- class [FacServiceAbstract](#)
- class [FacSupervisor](#)
- struct [FlightRequestStatus](#)
- class [AirInvServer](#)
- class [Connection](#)
- struct [header](#)
- struct [Reply](#)
- struct [Request](#)
- class [RequestHandler](#)  
*The common handler for all incoming requests.*



- class [RequestParser](#)  
*Parser for incoming requests.*
- class [AIRINV\\_Master\\_ServiceContext](#)
- class [AIRINV\\_ServiceContext](#)  
*Class holding the context of the AirInv services.*
- class [ServiceAbstract](#)

## Namespaces

- namespace [FFDisutilityParserHelper](#)
- namespace [FRAT5ParserHelper](#)
- namespace [InventoryParserHelper](#)
- namespace [ScheduleParserHelper](#)
- namespace [DCPPParserHelper](#)

## Typedefs

- typedef boost::shared\_ptr< [AIRINV\\_Service](#) > [AIRINV\\_ServicePtr\\_T](#)
- typedef boost::shared\_ptr< [AIRINV\\_Master\\_Service](#) > [AIRINV\\_Master\\_ServicePtr\\_T](#)
- typedef std::map< const stdair::AirlineCode\_T, [AIRINV\\_ServicePtr\\_T](#) > [AIRINV\\_ServicePtr\\_Map\\_T](#)
- typedef std::map< const stdair::DTD\_T, double > [FRAT5Curve\\_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, 1, 2 > [uint1\\_2\\_p\\_t](#)
- typedef boost::spirit::classic::uint\_parser< unsigned int, 10, 1, 3 > [uint1\\_3\\_p\\_t](#)
- typedef boost::spirit::classic::uint\_parser< unsigned int, 10, 4, 4 > [uint4\\_p\\_t](#)
- typedef boost::spirit::classic::uint\_parser< unsigned int, 10, 1, 4 > [uint1\\_4\\_p\\_t](#)
- typedef boost::spirit::classic::chset< [char\\_t](#) > [chset\\_t](#)
- typedef boost::spirit::classic::impl::loop\_traits< [chset\\_t](#), unsigned int, unsigned int >::type [repeat\\_p\\_t](#)
- typedef boost::spirit::classic::bounded< [uint2\\_p\\_t](#), unsigned int > [bounded2\\_p\\_t](#)
- typedef boost::spirit::classic::bounded< [uint1\\_2\\_p\\_t](#), unsigned int > [bounded1\\_2\\_p\\_t](#)
- typedef boost::spirit::classic::bounded< [uint1\\_3\\_p\\_t](#), unsigned int > [bounded1\\_3\\_p\\_t](#)
- typedef boost::spirit::classic::bounded< [uint4\\_p\\_t](#), unsigned int > [bounded4\\_p\\_t](#)
- typedef boost::spirit::classic::bounded< [uint1\\_4\\_p\\_t](#), unsigned int > [bounded1\\_4\\_p\\_t](#)
- typedef std::set< stdair::AirportCode\_T > [AirportList\\_T](#)
- typedef std::vector< stdair::AirportCode\_T > [AirportOrderedList\\_T](#)
- typedef std::vector< [BookingClassStruct](#) > [BookingClassStructList\\_T](#)
- typedef std::vector< [BucketStruct](#) > [BucketStructList\\_T](#)
- typedef std::vector< [FareFamilyStruct](#) > [FareFamilyStructList\\_T](#)
- typedef std::vector< [LegCabinStruct](#) > [LegCabinStructList\\_T](#)
- typedef std::vector< [LegStruct](#) > [LegStructList\\_T](#)

- typedef std::vector< [SegmentCabinStruct](#) > [SegmentCabinStructList\\_T](#)
- typedef std::vector< [SegmentStruct](#) > [SegmentStructList\\_T](#)
- typedef std::map< const stdair::Date\_T, stdair::SegmentCabin \* > [DepartureDateSegmentCabinMap\\_T](#)
- typedef std::map< const std::string, [DepartureDateSegmentCabinMap\\_T](#) > [SimilarSegmentCabinSetMap\\_T](#)
- typedef boost::shared\_ptr< boost::thread > [ThreadShrPtr\\_T](#)
- typedef std::vector< [ThreadShrPtr\\_T](#) > [ThreadShrPtrList\\_T](#)
- typedef boost::shared\_ptr< [Connection](#) > [ConnectionShrPtr\\_T](#)

## Variables

- const std::string [DEFAULT\\_AIRLINE\\_CODE](#) = "BA"
- const [FRAT5Curve\\_T](#) [DEFAULT\\_PICKUP\\_FRAT5\\_CURVE](#)
- const std::string [DEFAULT\\_AIRLINE\\_CODE](#)
- const [FRAT5Curve\\_T](#) [DEFAULT\\_PICKUP\\_FRAT5\\_CURVE](#)

### 9.1.1 Typedef Documentation

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

(Smart) Pointer on the AirInv (slave) service handler.

Definition at line 210 of file [AIRINV\\_Types.hpp](#).

#### 9.1.1.2 typedef boost::shared\_ptr<[AIRINV\\_Master\\_Service](#)> [AIRINV::AIRINV\\_Master\\_ServicePtr\\_T](#)

(Smart) Pointer on the AirInv master service handler.

Definition at line 215 of file [AIRINV\\_Types.hpp](#).

#### 9.1.1.3 typedef std::map<const stdair::AirlineCode\_T, [AIRINV\\_ServicePtr\\_T](#)> [AIRINV::AIRINV\\_ServicePtr\\_Map\\_T](#)

Type defining a map of airline codes and the corresponding airline inventories.

Definition at line 222 of file [AIRINV\\_Types.hpp](#).

#### 9.1.1.4 typedef std::map<const stdair::DTD\_T, double> [AIRINV::FRAT5Curve\\_T](#)

Define the FRAT5 curve.

Definition at line 227 of file [AIRINV\\_Types.hpp](#).

#### 9.1.1.5 typedef char [AIRINV::char\\_t](#)

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

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

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

**9.1.1.7** `typedef boost::spirit::classic::scanner<iterator_t> AIRINV::scanner_t`

Definition at line 36 of file BasParserTypes.hpp.

**9.1.1.8** `typedef boost::spirit::classic::rule<scanner_t> AIRINV::rule_t`

Definition at line 37 of file BasParserTypes.hpp.

**9.1.1.9** `typedef boost::spirit::classic::int_parser<unsigned int, 10, 1, 1> AIRINV::int1_p_t`

1-digit-integer parser

Definition at line 45 of file BasParserTypes.hpp.

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

2-digit-integer parser

Definition at line 48 of file BasParserTypes.hpp.

**9.1.1.11** `typedef boost::spirit::classic::uint_parser<unsigned int, 10, 1, 2> AIRINV::uint1_2_p_t`

Up-to-2-digit-integer parser

Definition at line 51 of file BasParserTypes.hpp.

**9.1.1.12** `typedef boost::spirit::classic::uint_parser<unsigned int, 10, 1, 3> AIRINV::uint1_3_p_t`

Up-to-3-digit-integer parser

Definition at line 54 of file BasParserTypes.hpp.

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

4-digit-integer parser

Definition at line 57 of file BasParserTypes.hpp.

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

Up-to-4-digit-integer parser

Definition at line 60 of file BasParserTypes.hpp.

**9.1.1.15** `typedef boost::spirit::classic::chset<char_t> AIRINV::chset_t`

character set

Definition at line 63 of file BasParserTypes.hpp.

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

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

Definition at line 69 of file BasParserTypes.hpp.

**9.1.1.17** `typedef boost::spirit::classic::bounded<uint2_p_t, unsigned int> AIRINV::bounded2_p_t`

Bounded-number-of-integers parser

Definition at line 72 of file BasParserTypes.hpp.

**9.1.1.18** `typedef boost::spirit::classic::bounded<uint1_2_p_t, unsigned int> AIRINV::bounded1_2_p_t`

Definition at line 73 of file BasParserTypes.hpp.

**9.1.1.19** `typedef boost::spirit::classic::bounded<uint1_3_p_t, unsigned int> AIRINV::bounded1_3_p_t`

Definition at line 74 of file BasParserTypes.hpp.

**9.1.1.20** `typedef boost::spirit::classic::bounded<uint4_p_t, unsigned int> AIRINV::bounded4_p_t`

Definition at line 75 of file BasParserTypes.hpp.

**9.1.1.21** `typedef boost::spirit::classic::bounded<uint1_4_p_t, unsigned int> AIRINV::bounded1_4_p_t`

Definition at line 76 of file BasParserTypes.hpp.

**9.1.1.22** `typedef std::set<stdair::AirportCode_T> AIRINV::AirportList_T`

Define lists of Airport Codes.

Definition at line 16 of file AirportList.hpp.

**9.1.1.23** `typedef std::vector<stdair::AirportCode_T> AIRINV::AirportOrderedList_T`

Definition at line 17 of file AirportList.hpp.

**9.1.1.24** `typedef std::vector<BookingClassStruct> AIRINV::BookingClassStructList_T`

List of BookingClass structures.

Definition at line 60 of file BookingClassStruct.hpp.

**9.1.1.25** `typedef std::vector<BucketStruct> AIRINV::BucketStructList_T`

List of Bucket structures.

Definition at line 44 of file BucketStruct.hpp.

**9.1.1.26** `typedef std::vector<FareFamilyStruct> AIRINV::FareFamilyStructList_T`

List of FareFamily-Detail structures.

Definition at line 59 of file FareFamilyStruct.hpp.

**9.1.1.27** `typedef std::vector<LegCabinStruct> AIRINV::LegCabinStructList_T`

List of LegCabin-Detail structures.

Definition at line 52 of file LegCabinStruct.hpp.

**9.1.1.28** `typedef std::vector<LegStruct> AIRINV::LegStructList_T`

List of Leg structures.

Definition at line 57 of file LegStruct.hpp.

**9.1.1.29** `typedef std::vector<SegmentCabinStruct> AIRINV::SegmentCabinStructList_T`

List of SegmentCabin-Detail structures.

Definition at line 48 of file SegmentCabinStruct.hpp.

**9.1.1.30** `typedef std::vector<SegmentStruct> AIRINV::SegmentStructList_T`

List of Segment structures.

Definition at line 43 of file SegmentStruct.hpp.

**9.1.1.31** `typedef std::map<const stdair::Date_T, stdair::SegmentCabin*> AIRINV::DepartureDateSegmentCabinMap\_T`

Definition at line 31 of file InventoryManager.hpp.

**9.1.1.32** `typedef std::map<const std::string, DepartureDateSegmentCabinMap\_T> AIRINV::SimilarSegmentCabinSetMap\_T`

Definition at line 33 of file InventoryManager.hpp.

**9.1.1.33** `typedef boost::shared_ptr<boost::thread> AIRINV::ThreadShrPtr_T`

Definition at line 15 of file AirInvServer\_ASIO.cpp.

**9.1.1.34** `typedef std::vector<ThreadShrPtr\_T> AIRINV::ThreadShrPtrList_T`

Definition at line 16 of file AirInvServer\_ASIO.cpp.

**9.1.1.35** `typedef boost::shared_ptr<Connection> AIRINV::ConnectionShrPtr_T`

Shared pointer on a [Connection](#) object.

Definition at line 71 of file Connection.hpp.

### 9.1.2 Variable Documentation

#### 9.1.2.1 `const std::string AIRINV::DEFAULT_AIRLINE_CODE = "BA"`

Default airline name for the [AIRINV\\_Service](#).

Definition at line 11 of file BasConst.cpp.

#### 9.1.2.2 `const FRAT5Curve_T AIRINV::DEFAULT_PICKUP_FRAT5_CURVE`

**Initial value:**

```
DefaultMap::createPickupFRAT5Curve()
```

Default pick-up FRAT5 curve for Q-equivalent booking conversion.

Definition at line 14 of file BasConst.cpp.

#### 9.1.2.3 `const std::string AIRINV::DEFAULT_AIRLINE_CODE`

Default airline name for the [AIRINV\\_Service](#).

Definition at line 11 of file BasConst.cpp.

#### 9.1.2.4 `const FRAT5Curve_T AIRINV::DEFAULT_PICKUP_FRAT5_CURVE`

Default pick-up FRAT5 curve for Q-equivalent booking conversion.

Definition at line 14 of file BasConst.cpp.

## 9.2 AIRINV::DCPParserHelper Namespace Reference

### Classes

- struct [ParserSemanticAction](#)
- struct [storeDCPIId](#)
- struct [storeOrigin](#)
- struct [storeDestination](#)
- struct [storeDateRangeStart](#)
- struct [storeDateRangeEnd](#)
- struct [storeStartRangeTime](#)
- struct [storeEndRangeTime](#)
- struct [storePOS](#)
- struct [storeCabinCode](#)
- struct [storeChannel](#)
- struct [storeAdvancePurchase](#)
- struct [storeSaturdayStay](#)
- struct [storeChangeFees](#)
- struct [storeNonRefundable](#)
- struct [storeMinimumStay](#)
- struct [storeDCP](#)
- struct [storeAirlineCode](#)
- struct [storeClass](#)
- struct [doEndDCP](#)
- struct [DCPRuleParser](#)

## Variables

- [stdair::int1\\_p\\_t int1\\_p](#)
- [stdair::uint2\\_p\\_t uint2\\_p](#)
- [stdair::uint4\\_p\\_t uint4\\_p](#)
- [stdair::uint1\\_4\\_p\\_t uint1\\_4\\_p](#)
- [stdair::hour\\_p\\_t hour\\_p](#)
- [stdair::minute\\_p\\_t minute\\_p](#)
- [stdair::second\\_p\\_t second\\_p](#)
- [stdair::year\\_p\\_t year\\_p](#)
- [stdair::month\\_p\\_t month\\_p](#)
- [stdair::day\\_p\\_t day\\_p](#)

### 9.2.1 Variable Documentation

#### 9.2.1.1 [stdair::int1\\_p\\_t AIRINV::DCPParserHelper::int1\\_p](#)

1-digit-integer parser

Definition at line 427 of file DCPParserHelper.cpp.

#### 9.2.1.2 [stdair::uint2\\_p\\_t AIRINV::DCPParserHelper::uint2\\_p](#)

2-digit-integer parser

Definition at line 430 of file DCPParserHelper.cpp.

#### 9.2.1.3 [stdair::uint4\\_p\\_t AIRINV::DCPParserHelper::uint4\\_p](#)

4-digit-integer parser

Definition at line 433 of file DCPParserHelper.cpp.

#### 9.2.1.4 [stdair::uint1\\_4\\_p\\_t AIRINV::DCPParserHelper::uint1\\_4\\_p](#)

Up-to-4-digit-integer parser

Definition at line 436 of file DCPParserHelper.cpp.

Referenced by [AIRINV::DCPParserHelper::DCPRuleParser::DCPRuleParser\(\)](#).

#### 9.2.1.5 [stdair::hour\\_p\\_t AIRINV::DCPParserHelper::hour\\_p](#)

Time element parsers.

Definition at line 439 of file DCPParserHelper.cpp.

Referenced by [AIRINV::DCPParserHelper::DCPRuleParser::DCPRuleParser\(\)](#).

#### 9.2.1.6 [stdair::minute\\_p\\_t AIRINV::DCPParserHelper::minute\\_p](#)

Definition at line 440 of file DCPParserHelper.cpp.

Referenced by [AIRINV::DCPParserHelper::DCPRuleParser::DCPRuleParser\(\)](#).

**9.2.1.7 stdair::second\_p\_t AIRINV::DCPParserHelper::second\_p**

Definition at line 441 of file DCPParserHelper.cpp.

Referenced by AIRINV::DCPParserHelper::DCPRuleParser::DCPRuleParser().

**9.2.1.8 stdair::year\_p\_t AIRINV::DCPParserHelper::year\_p**

Date element parsers.

Definition at line 444 of file DCPParserHelper.cpp.

Referenced by AIRINV::DCPParserHelper::DCPRuleParser::DCPRuleParser().

**9.2.1.9 stdair::month\_p\_t AIRINV::DCPParserHelper::month\_p**

Definition at line 445 of file DCPParserHelper.cpp.

Referenced by AIRINV::DCPParserHelper::DCPRuleParser::DCPRuleParser().

**9.2.1.10 stdair::day\_p\_t AIRINV::DCPParserHelper::day\_p**

Definition at line 446 of file DCPParserHelper.cpp.

Referenced by AIRINV::DCPParserHelper::DCPRuleParser::DCPRuleParser().

**9.3 AIRINV::FFDisutilityParserHelper Namespace Reference****Classes**

- struct [ParserSemanticAction](#)
- struct [storeCurveKey](#)
- struct [storeDTD](#)
- struct [storeFFDisutilityValue](#)
- struct [doEndCurve](#)
- struct [FFDisutilityParser](#)

**Functions**

- [repeat\\_p\\_t key\\_p](#) ([chset\\_t](#)("0-9A-Z").derived(), 1, 10)

**9.3.1 Function Documentation****9.3.1.1 repeat\_p\_t AIRINV::FFDisutilityParserHelper::key\_p ([chset\\_t](#)("0-9A-Z").derived(), 1, 10)**

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

Referenced by AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::definition().



## 9.4 AIRINV::FRAT5ParserHelper Namespace Reference

### Classes

- struct [ParserSemanticAction](#)
- struct [storeCurveKey](#)
- struct [storeDTD](#)
- struct [storeFRAT5Value](#)
- struct [doEndCurve](#)
- struct [FRAT5Parser](#)

### Functions

- [repeat\\_p\\_t key\\_p](#) ([chset\\_t](#)("0-9A-Z").derived(), 1, 10)

#### 9.4.1 Function Documentation

##### 9.4.1.1 [repeat\\_p\\_t](#) AIRINV::FRAT5ParserHelper::key\_p ([chset\\_t](#)("0-9A-Z").derived(), 1, 10)

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

Referenced by AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::definition().

## 9.5 AIRINV::InventoryParserHelper Namespace Reference

### Classes

- struct [ParserSemanticAction](#)
- struct [storeSnapshotDate](#)
- struct [storeAirlineCode](#)
- struct [storeFlightNumber](#)
- struct [storeFlightDate](#)
- struct [storeFlightTypeCode](#)
- struct [storeFlightVisibilityCode](#)
- struct [storeLegBoardingPoint](#)
- struct [storeLegOffPoint](#)
- struct [storeOperatingAirlineCode](#)
- struct [storeOperatingFlightNumber](#)
- struct [storeBoardingDate](#)
- struct [storeBoardingTime](#)
- struct [storeOffDate](#)
- struct [storeOffTime](#)
- struct [storeLegCabinCode](#)
- struct [storeSaleableCapacity](#)
- struct [storeAU](#)
- struct [storeUPR](#)
- struct [storeBookingCounter](#)
- struct [storeNAV](#)
- struct [storeGAV](#)
- struct [storeACP](#)
- struct [storeETB](#)

- struct [storeYieldUpperRange](#)
- struct [storeBucketAvailality](#)
- struct [storeSeatIndex](#)
- struct [storeSegmentBoardingPoint](#)
- struct [storeSegmentOffPoint](#)
- struct [storeSegmentCabinCode](#)
- struct [storeSegmentCabinBookingCounter](#)
- struct [storeClassCode](#)
- struct [storeSubclassCode](#)
- struct [storeParentClassCode](#)
- struct [storeParentSubclassCode](#)
- struct [storeCumulatedProtection](#)
- struct [storeProtection](#)
- struct [storeNego](#)
- struct [storeNoShow](#)
- struct [storeOverbooking](#)
- struct [storeNbOfBkgs](#)
- struct [storeNbOfGroupBkgs](#)
- struct [storeNbOfPendingGroupBkgs](#)
- struct [storeNbOfStaffBkgs](#)
- struct [storeNbOfWLBkgs](#)
- struct [storeClassETB](#)
- struct [storeClassAvailability](#)
- struct [storeSegmentAvailability](#)
- struct [storeRevenueAvailability](#)
- struct [storeFamilyCode](#)
- struct [storeFClasses](#)
- struct [doEndFlightDate](#)
- struct [InventoryParser](#)

## 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)
- [bounded2\\_p\\_t year\\_p](#) (uint2\_p.derived(), 0u, 99u)
- [bounded2\\_p\\_t month\\_p](#) (uint2\_p.derived(), 1u, 12u)
- [bounded2\\_p\\_t day\\_p](#) (uint2\_p.derived(), 1u, 31u)
- [repeat\\_p\\_t dow\\_p](#) (chset\_t("0-1").derived().derived(), 7, 7)
- [repeat\\_p\\_t airport\\_p](#) (chset\_t("0-9A-Z").derived(), 3, 3)
- [bounded1\\_2\\_p\\_t hours\\_p](#) (uint1\_2\_p.derived(), 0u, 24u)
- [bounded2\\_p\\_t minutes\\_p](#) (uint2\_p.derived(), 0u, 59u)
- [bounded2\\_p\\_t seconds\\_p](#) (uint2\_p.derived(), 0u, 59u)
- [chset\\_t cabin\\_code\\_p](#) ("A-Z")
- [chset\\_t class\\_code\\_p](#) ("A-Z")
- [chset\\_t passenger\\_type\\_p](#) ("A-Z")
- [repeat\\_p\\_t class\\_code\\_list\\_p](#) (chset\_t("A-Z").derived(), 1, 26)
- [bounded1\\_3\\_p\\_t stay\\_duration\\_p](#) (uint1\_3\_p.derived(), 0u, 999u)

## Variables

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

### 9.5.1 Function Documentation

#### 9.5.1.1 [repeat\\_p\\_t](#) AIRINV::InventoryParserHelper::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 AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

#### 9.5.1.2 [bounded1\\_4\\_p\\_t](#) AIRINV::InventoryParserHelper::flight\_number\_p (uint1\_4\_p. derived(), 0u, 9999u)

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

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

#### 9.5.1.3 [bounded2\\_p\\_t](#) AIRINV::InventoryParserHelper::year\_p (uint2\_p. derived(), 0u, 99u)

Year Parser: limit\_d(00u, 99u)[uint4\_p]

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

#### 9.5.1.4 [bounded2\\_p\\_t](#) AIRINV::InventoryParserHelper::month\_p (uint2\_p. derived(), 1u, 12u)

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

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

#### 9.5.1.5 [bounded2\\_p\\_t](#) AIRINV::InventoryParserHelper::day\_p (uint2\_p. derived(), 1u, 31u)

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

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

#### 9.5.1.6 [repeat\\_p\\_t](#) AIRINV::InventoryParserHelper::dow\_p ([chset\\_t](#)("0-1").derived().derived(), 7, 7)

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

#### 9.5.1.7 [repeat\\_p\\_t](#) AIRINV::InventoryParserHelper::airport\_p ([chset\\_t](#)("0-9A-Z").derived(), 3, 3)

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

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**9.5.1.8 bounded1\_2\_p\_t AIRINV::InventoryParserHelper::hours\_p (uint1\_2\_p. *derived()*, 0u, 24u)**

Hour Parser: limit\_d(0u, 24u)[uint2\_p]

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**9.5.1.9 bounded2\_p\_t AIRINV::InventoryParserHelper::minutes\_p (uint2\_p. *derived()*, 0u, 59u)**

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

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**9.5.1.10 bounded2\_p\_t AIRINV::InventoryParserHelper::seconds\_p (uint2\_p. *derived()*, 0u, 59u)**

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

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**9.5.1.11 chset\_t AIRINV::InventoryParserHelper::cabin\_code\_p ("A-Z")**

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

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**9.5.1.12 chset\_t AIRINV::InventoryParserHelper::class\_code\_p ("A-Z")**

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

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**9.5.1.13 chset\_t AIRINV::InventoryParserHelper::passenger\_type\_p ("A-Z")**

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

**9.5.1.14 repeat\_p\_t AIRINV::InventoryParserHelper::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 AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**9.5.1.15 bounded1\_3\_p\_t AIRINV::InventoryParserHelper::stay\_duration\_p (uint1\_3\_p. *derived()*, 0u, 999u)**

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

**9.5.2 Variable Documentation****9.5.2.1 int1\_p\_t AIRINV::InventoryParserHelper::int1\_p**

1-digit-integer parser

Definition at line 823 of file InventoryParserHelper.cpp.

### 9.5.2.2 [uint2\\_p\\_t AIRINV::InventoryParserHelper::uint2\\_p](#)

2-digit-integer parser

Definition at line 826 of file InventoryParserHelper.cpp.

### 9.5.2.3 [uint1\\_2\\_p\\_t AIRINV::InventoryParserHelper::uint1\\_2\\_p](#)

Up-to-2-digit-integer parser

Definition at line 829 of file InventoryParserHelper.cpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

### 9.5.2.4 [uint1\\_3\\_p\\_t AIRINV::InventoryParserHelper::uint1\\_3\\_p](#)

Up-to-3-digit-integer parser

Definition at line 832 of file InventoryParserHelper.cpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

### 9.5.2.5 [uint4\\_p\\_t AIRINV::InventoryParserHelper::uint4\\_p](#)

4-digit-integer parser

Definition at line 835 of file InventoryParserHelper.cpp.

### 9.5.2.6 [uint1\\_4\\_p\\_t AIRINV::InventoryParserHelper::uint1\\_4\\_p](#)

Up-to-4-digit-integer parser

Definition at line 838 of file InventoryParserHelper.cpp.

### 9.5.2.7 [int1\\_p\\_t AIRINV::InventoryParserHelper::family\\_code\\_p](#)

Family code parser

Definition at line 880 of file InventoryParserHelper.cpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

## 9.6 AIRINV::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](#)

### 9.6.1 Function Documentation

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

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

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

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

**9.6.1.3 [bounded4\\_p\\_t](#) AIRINV::ScheduleParserHelper::year\_p (uint4\_p. *derived()*, 2000u, 2099u)**

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

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

**9.6.1.4 [bounded2\\_p\\_t](#) AIRINV::ScheduleParserHelper::month\_p (uint2\_p. *derived()*, 1u, 12u)**

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

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

**9.6.1.5 [bounded2\\_p\\_t](#) AIRINV::ScheduleParserHelper::day\_p (uint2\_p. *derived()*, 1u, 31u)**

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

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

**9.6.1.6 [repeat\\_p\\_t](#) AIRINV::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 AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition().

**9.6.1.7 [repeat\\_p\\_t](#) AIRINV::ScheduleParserHelper::airport\_p ([chset\\_t](#)("0-9A-Z").*derived()*, 3, 3)**

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

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

**9.6.1.8 [bounded2\\_p\\_t](#) AIRINV::ScheduleParserHelper::hours\_p (uint2\_p. *derived()*, 0u, 23u)**

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

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

**9.6.1.9 [bounded2\\_p\\_t](#) AIRINV::ScheduleParserHelper::minutes\_p (uint2\_p. *derived()*, 0u, 59u)**

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

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

#### 9.6.1.10 [bounded2\\_p\\_t](#) AIRINV::ScheduleParserHelper::seconds\_p (uint2\_p. *derived()*, 0u, 59u)

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

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

#### 9.6.1.11 [chset\\_t](#) AIRINV::ScheduleParserHelper::cabin\_code\_p ("A-Z")

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

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

#### 9.6.1.12 [repeat\\_p\\_t](#) AIRINV::ScheduleParserHelper::key\_p ([chset\\_t](#)("0-9A-Z").*derived()*, 1, 10)

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

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

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

### 9.6.2 Variable Documentation

#### 9.6.2.1 [int1\\_p\\_t](#) AIRINV::ScheduleParserHelper::int1\_p

1-digit-integer parser

Definition at line 469 of file ScheduleParserHelper.cpp.

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

#### 9.6.2.2 [uint2\\_p\\_t](#) AIRINV::ScheduleParserHelper::uint2\_p

2-digit-integer parser

Definition at line 472 of file ScheduleParserHelper.cpp.

#### 9.6.2.3 [uint4\\_p\\_t](#) AIRINV::ScheduleParserHelper::uint4\_p

4-digit-integer parser

Definition at line 475 of file ScheduleParserHelper.cpp.



**9.6.2.4 uint1\_4\_p\_t AIRINV::ScheduleParserHelper::uint1\_4\_p**

Up-to-4-digit-integer parser

Definition at line 478 of file ScheduleParserHelper.cpp.

**9.6.2.5 int1\_p\_t AIRINV::ScheduleParserHelper::family\_code\_p**

Family code parser

Definition at line 514 of file ScheduleParserHelper.cpp.

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

**9.7 stdair Namespace Reference**

Forward declarations.

**Classes**

- struct [BomPropertyTree](#)

**9.7.1 Detailed Description**

Forward declarations.

**10 AirInv Class Documentation****10.1 AIRINV::AIRINV\_Master\_Service Class Reference**

Interface for the [AIRINV](#) Services.

```
#include <airinv/AIRINV_Master_Service.hpp>
```

**Public Member Functions**

- [AIRINV\\_Master\\_Service](#) (const stdair::BasLogParams &, const stdair::BasDBParams &)
- [AIRINV\\_Master\\_Service](#) (const stdair::BasLogParams &)
- [AIRINV\\_Master\\_Service](#) (stdair::STDAIR\_ServicePtr\_T)
- [AIRINV\\_Master\\_Service](#) (stdair::STDAIR\_ServicePtr\_T, SEVMGR::SEVMGR\_ServicePtr\_T)
- void [parseAndLoad](#) (const [InventoryFilePath](#) &)
- void [parseAndLoad](#) (const stdair::ScheduleFilePath &, const stdair::ODFilePath &, const stdair::FRAT5FilePath &, const stdair::FFDisutilityFilePath &, const AIRRAC::YieldFilePath &)
- [~AIRINV\\_Master\\_Service](#) ()
- void [initSnapshotAndRMEvents](#) (const stdair::Date\_T &, const stdair::Date\_T &)
- void [buildSampleBom](#) ()
- void [clonePersistentBom](#) ()
- void [buildComplementaryLinks](#) (stdair::BomRoot &)
- void [calculateAvailability](#) (stdair::TravelSolutionStruct &)

- bool [sell](#) (const std::string &iSegmentDateKey, const stdair::ClassCode\_T &, const stdair::PartySize\_T &)
- bool [sell](#) (const stdair::BookingClassID\_T &, const stdair::PartySize\_T &)
- bool [cancel](#) (const std::string &iSegmentDateKey, const stdair::ClassCode\_T &, const stdair::PartySize\_T &)
- bool [cancel](#) (const stdair::BookingClassID\_T &, const stdair::PartySize\_T &)
- void [takeSnapshots](#) (const stdair::SnapshotStruct &)
- void [optimise](#) (const stdair::RMEventStruct &)
- std::string [jsonHandler](#) (const stdair::JSONString &) const
- std::string [jsonExportFlightDateList](#) (const stdair::AirlineCode\_T &iAirlineCode="all", const stdair::FlightNumber\_T &iFlightNumber=0) const
- std::string [jsonExportFlightDateObjects](#) (const stdair::AirlineCode\_T &, const stdair::FlightNumber\_T &, const stdair::Date\_T &iDepartureDate) const
- std::string [list](#) (const stdair::AirlineCode\_T &iAirlineCode="all", const stdair::FlightNumber\_T &iFlightNumber=0) const
- bool [check](#) (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

### 10.1.1 Detailed Description

Interface for the [AIRINV](#) Services.

Definition at line 47 of file AIRINV\_Master\_Service.hpp.

### 10.1.2 Constructor & Destructor Documentation

#### 10.1.2.1 AIRINV::AIRINV\_Master\_Service::AIRINV\_Master\_Service (const stdair::BasLogParams &, const stdair::BasDBParams &)

Constructor.

The initSlaveAirinvService() 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 46 of file AIRINV\_Master\_Service.cpp.

#### 10.1.2.2 AIRINV::AIRINV\_Master\_Service::AIRINV\_Master\_Service (const stdair::BasLogParams &)

Constructor.

The initSlaveAirinvService() 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 68 of file AIRINV\_Master\_Service.cpp.

**10.1.2.3 AIRINV::AIRINV\_Master\_Service::AIRINV\_Master\_Service (stdair::STDAIR\_ServicePtr\_T)**

Constructor.

The initSlaveAirinvService() 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 [AIRINV\\_Master\\_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 89 of file AIRINV\_Master\_Service.cpp.

**10.1.2.4 AIRINV::AIRINV\_Master\_Service::AIRINV\_Master\_Service (stdair::STDAIR\_ServicePtr\_T, SEVMGR::SEVMGR\_ServicePtr\_T)**

Constructor.

The initSlaveAirinvService() 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 [AIRINV\\_Master\\_Service](#) is itself being initialised by another library service such as SIMCRS\_Service).

**Parameters:**

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

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

Definition at line 106 of file AIRINV\_Master\_Service.cpp.

**10.1.2.5 AIRINV::AIRINV\_Master\_Service::~~AIRINV\_Master\_Service ()**

Destructor.

Definition at line 127 of file AIRINV\_Master\_Service.cpp.

**10.1.3 Member Function Documentation****10.1.3.1 void AIRINV::AIRINV\_Master\_Service::parseAndLoad (const [InventoryFilePath](#) &)**

Parse the inventory dump and load it into memory.

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

**Parameters:**

**const** [InventoryFilePath](#)& Filename of the input inventory file.

Definition at line 255 of file AIRINV\_Master\_Service.cpp.

References [clonePersistentBom\(\)](#), [AIRINV::AIRINV\\_Master\\_ServiceContext::getAIRINV\\_Service\(\)](#), [AIRINV::AIRINV\\_Master\\_ServiceContext::getOwnStdairServiceFlag\(\)](#), and [AIRINV::AIRINV\\_Service::parseAndLoad\(\)](#).

### 10.1.3.2 void AIRINV::AIRINV\_Master\_Service::parseAndLoad (const stdair::ScheduleFilePath &, const stdair::ODFilePath &, const stdair::FRAT5FilePath &, const stdair::FFDisutilityFilePath &, const AIRRAC::YieldFilePath &)

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.

**const** [stdair::FRAT5FilePath](#)& Filename of the input FRAT5 file.

**const** [stdair::FFDisutilityFilePath](#)& Filename of the input FF disutility file.

**const** [AIRRAC::YieldFilePath](#)& Filename of the input yield file.

Definition at line 353 of file AIRINV\_Master\_Service.cpp.

References [buildComplementaryLinks\(\)](#), [clonePersistentBom\(\)](#), [AIRINV::AIRINV\\_Master\\_ServiceContext::getAIRINV\\_Service\(\)](#), [AIRINV::AIRINV\\_Master\\_ServiceContext::getOwnStdairServiceFlag\(\)](#), [AIRINV::AIRINV\\_Master\\_ServiceContext::getSTDAIR\\_Service\(\)](#), and [AIRINV::AIRINV\\_Service::parseAndLoad\(\)](#).

### 10.1.3.3 void AIRINV::AIRINV\_Master\_Service::initSnapshotAndRMEvents (const stdair::Date\_T &, const stdair::Date\_T &)

Initialise the snapshot and RM events for the inventories.

**Parameters:**

**const** [stdair::Date\\_T](#)& Parameters for the start date.

**const** [stdair::Date\\_T](#)& Parameters for the end date.

Definition at line 647 of file AIRINV\_Master\_Service.cpp.

References [AIRINV::AIRINV\\_Master\\_ServiceContext::getAIRINV\\_Service\(\)](#), [AIRINV::AIRINV\\_Master\\_ServiceContext::getSEVMGR\\_ServicePtr\(\)](#), and [AIRINV::AIRINV\\_Service::initRMEvents\(\)](#).

### 10.1.3.4 void AIRINV::AIRINV\_Master\_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 291 of file AIRINV\_Master\_Service.cpp.

References buildComplementaryLinks(), AIRINV::AIRINV\_Service::buildSampleBom(), clonePersistentBom(), AIRINV::AIRINV\_Master\_ServiceContext::getAIRINV\_Service(), AIRINV::AIRINV\_Master\_ServiceContext::getOwnStdairServiceFlag(), and AIRINV::AIRINV\_Master\_ServiceContext::getSTDAIR\_Service().

#### 10.1.3.5 void AIRINV::AIRINV\_Master\_Service::clonePersistentBom ()

Clone the persistent BOM object.

Definition at line 415 of file AIRINV\_Master\_Service.cpp.

References buildComplementaryLinks(), AIRINV::AIRINV\_Service::clonePersistentBom(), AIRINV::AIRINV\_Master\_ServiceContext::getAIRINV\_Service(), AIRINV::AIRINV\_Master\_ServiceContext::getOwnStdairServiceFlag(), and AIRINV::AIRINV\_Master\_ServiceContext::getSTDAIR\_Service().

Referenced by buildSampleBom(), and parseAndLoad().

#### 10.1.3.6 void AIRINV::AIRINV\_Master\_Service::buildComplementaryLinks (stdair::BomRoot &)

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

##### Note:

Do nothing for now.

Definition at line 466 of file AIRINV\_Master\_Service.cpp.

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

#### 10.1.3.7 void AIRINV::AIRINV\_Master\_Service::calculateAvailability (stdair::TravelSolution-Struct &)

Compute the availability for the given travel solution.

Definition at line 684 of file AIRINV\_Master\_Service.cpp.

References AIRINV::AIRINV\_Service::calculateAvailability(), and AIRINV::AIRINV\_Master\_ServiceContext::getAIRINV\_Service().

#### 10.1.3.8 bool AIRINV::AIRINV\_Master\_Service::sell (const std::string & *iSegmentDateKey*, const stdair::ClassCode\_T &, const stdair::PartySize\_T &)

Register a booking.

##### Parameters:

**const** std::string& Key for the segment on which the sale is made.

**const** stdair::ClassCode\_T& Class code where the sale is made.

**const** stdair::PartySize\_T& Party size.

##### Returns:

bool Whether or not the sale was successful

Definition at line 714 of file AIRINV\_Master\_Service.cpp.

References AIRINV::AIRINV\_Master\_ServiceContext::getAIRINV\_Service(), and AIRINV::AIRINV\_Service::sell().

#### 10.1.3.9 bool AIRINV::AIRINV\_Master\_Service::sell (const stdair::BookingClassID\_T &, const stdair::PartySize\_T &)

Register a booking.

##### Parameters:

*const* stdair::BookingClassID\_T&  
*const* stdair::PartySize\_T& Party size.

##### Returns:

bool Whether or not the sale was successfull

Definition at line 756 of file AIRINV\_Master\_Service.cpp.

References AIRINV::AIRINV\_Master\_ServiceContext::getAIRINV\_Service(), and AIRINV::AIRINV\_Service::sell().

#### 10.1.3.10 bool AIRINV::AIRINV\_Master\_Service::cancel (const std::string & iSegmentDateKey, const stdair::ClassCode\_T &, const stdair::PartySize\_T &)

Register a cancellation.

##### Parameters:

*const* std::string& Key for the segment on which the cancellation is made.  
*const* stdair::ClassCode\_T& Class code where the sale is made.  
*const* stdair::PartySize\_T& Party size.

##### Returns:

bool Whether or not the cancellation was successfull

Definition at line 797 of file AIRINV\_Master\_Service.cpp.

References AIRINV::AIRINV\_Service::cancel(), and AIRINV::AIRINV\_Master\_ServiceContext::getAIRINV\_Service().

#### 10.1.3.11 bool AIRINV::AIRINV\_Master\_Service::cancel (const stdair::BookingClassID\_T &, const stdair::PartySize\_T &)

Register a cancellation.

##### Parameters:

*stdair::BookingClassID\_T&*  
*const* stdair::PartySize\_T& Party size.

##### Returns:

bool Whether or not the cancellation was successfull

Definition at line 839 of file AIRINV\_Master\_Service.cpp.

References AIRINV::AIRINV\_Service::cancel(), and AIRINV::AIRINV\_Master\_ServiceContext::getAIRINV\_Service().

#### 10.1.3.12 void AIRINV::AIRINV\_Master\_Service::takeSnapshots (const stdair::SnapshotStruct &)

Take inventory snapshots.

Definition at line 881 of file AIRINV\_Master\_Service.cpp.

References AIRINV::AIRINV\_Master\_ServiceContext::getAIRINV\_Service(), and AIRINV::AIRINV\_Service::takeSnapshots().

#### 10.1.3.13 void AIRINV::AIRINV\_Master\_Service::optimise (const stdair::RMEventStruct &)

Optimise (revenue management) an flight-date/network-date

Definition at line 907 of file AIRINV\_Master\_Service.cpp.

References AIRINV::AIRINV\_Master\_ServiceContext::getAIRINV\_Service(), and AIRINV::AIRINV\_Service::optimise().

#### 10.1.3.14 std::string AIRINV::AIRINV\_Master\_Service::jsonHandler (const stdair::JSONString &) const

Dispatch the JSon command string to the right JSon Service, according to the JSon command type.

##### Parameters:

**const** stdair::JSONString& Input string which contained the JSon command string.

##### Returns:

std::string Output string in which the asking objects are logged/dumped in a JSon format.

Definition at line 472 of file AIRINV\_Master\_Service.cpp.

References AIRINV::AIRINV\_Master\_ServiceContext::getAIRINV\_Service(), and AIRINV::AIRINV\_Service::jsonHandler().

#### 10.1.3.15 std::string AIRINV::AIRINV\_Master\_Service::jsonExportFlightDateList (const stdair::AirlineCode\_T & iAirlineCode = "all", const stdair::FlightNumber\_T & iFlightNumber = 0) const

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

##### Parameters:

**const** AirlineCode& Airline for which the flight-dates should be displayed. If set to "all" (default), all the inventories will be displayed.

**const** FlightNumber\_T& Flight number for which all the departure dates should be displayed. If set to 0 (the default), all the flight numbers will be displayed.

Definition at line 496 of file AIRINV\_Master\_Service.cpp.

References AIRINV::AIRINV\_Master\_ServiceContext::getAIRINV\_Service(), and AIRINV::AIRINV\_Service::jsonExportFlightDateList().

**10.1.3.16** `std::string AIRINV::AIRINV_Master_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-date 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 the flight to dump.

**Returns:**

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

Definition at line 521 of file AIRINV\_Master\_Service.cpp.

References AIRINV::AIRINV\_Master\_ServiceContext::getAIRINV\_Service(), and AIRINV::AIRINV\_Service::jsonExportFlightDateObjects().

**10.1.3.17** `std::string AIRINV::AIRINV_Master_Service::list (const stdair::AirlineCode_T & iAirlineCode = "all", const stdair::FlightNumber_T & iFlightNumber = 0) const`

Display the list of flight-dates (contained within the BOM tree) corresponding to the parameters given as input.

**Parameters:**

- const* AirlineCode& Airline for which the flight-dates should be displayed. If set to "all" (the default), all the inventories will be displayed.
- const* FlightNumber\_T& Flight number for which all the departure dates should be displayed. If set to 0 (the default), all the flight numbers will be displayed.

**Returns:**

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

Definition at line 548 of file AIRINV\_Master\_Service.cpp.

References AIRINV::AIRINV\_Master\_ServiceContext::getAIRINV\_Service(), and AIRINV::AIRINV\_Service::list().

**10.1.3.18** `bool AIRINV::AIRINV_Master_Service::check (const stdair::AirlineCode_T &, const stdair::FlightNumber_T &, const stdair::Date_T & iDepartureDate) const`

Check whether the given flight-date is a valid one.

**Parameters:**

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



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

**const** stdair::Date\_T& Departure date of the flight to check.

#### Returns:

bool Whether or not the given flight date is valid.

Definition at line 573 of file AIRINV\_Master\_Service.cpp.

References AIRINV::AIRINV\_Service::check(), and AIRINV::AIRINV\_Master\_ServiceContext::getAIRINV\_Service().

#### 10.1.3.19 std::string AIRINV::AIRINV\_Master\_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 598 of file AIRINV\_Master\_Service.cpp.

References AIRINV::AIRINV\_Service::csvDisplay(), and AIRINV::AIRINV\_Master\_ServiceContext::getAIRINV\_Service().

#### 10.1.3.20 std::string AIRINV::AIRINV\_Master\_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-date corresponding to the parameters given as input.

#### Parameters:

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

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

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

#### Returns:

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

Definition at line 621 of file AIRINV\_Master\_Service.cpp.

References AIRINV::AIRINV\_Service::csvDisplay(), and AIRINV::AIRINV\_Master\_ServiceContext::getAIRINV\_Service().

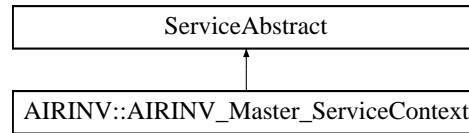
The documentation for this class was generated from the following files:

- [airinv/AIRINV\\_Master\\_Service.hpp](#)
- [airinv/service/AIRINV\\_Master\\_Service.cpp](#)

## 10.2 AIRINV::AIRINV\_Master\_ServiceContext Class Reference

```
#include <airinv/service/AIRINV_Master_ServiceContext.hpp>
```

Inheritance diagram for AIRINV::AIRINV\_Master\_ServiceContext::



## Friends

- class [AIRINV\\_Master\\_Service](#)
- class [FacAirinvMasterServiceContext](#)

### 10.2.1 Detailed Description

Class holding the context of the Airinv services.

Definition at line 28 of file AIRINV\_Master\_ServiceContext.hpp.

### 10.2.2 Friends And Related Function Documentation

#### 10.2.2.1 friend class [AIRINV\\_Master\\_Service](#) [friend]

The [AIRINV\\_Master\\_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 34 of file AIRINV\_Master\_ServiceContext.hpp.

#### 10.2.2.2 friend class [FacAirinvMasterServiceContext](#) [friend]

Definition at line 35 of file AIRINV\_Master\_ServiceContext.hpp.

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

- [airinv/service/AIRINV\\_Master\\_ServiceContext.hpp](#)
- [airinv/service/AIRINV\\_Master\\_ServiceContext.cpp](#)

## 10.3 AIRINV::AIRINV\_Service Class Reference

Interface for the [AIRINV](#) Services.

```
#include <airinv/AIRINV_Service.hpp>
```

### Public Member Functions

- [AIRINV\\_Service](#) (const stdair::BasLogParams &, const stdair::BasDBParams &)
- [AIRINV\\_Service](#) (const stdair::BasLogParams &)
- [AIRINV\\_Service](#) (stdair::STDAIR\_ServicePtr\_T)
- [AIRINV\\_Service](#) (stdair::STDAIR\_ServicePtr\_T, SEVMGR::SEVMGR\_ServicePtr\_T)
- void [parseAndLoad](#) (const [AIRINV::InventoryFilePath](#) &)

- void [parseAndLoad](#) (const stdair::ScheduleFilePath &, const stdair::ODFilePath &, const stdair::FRAT5FilePath &, const stdair::FFDisutilityFilePath &, const AIRRAC::YieldFilePath &)
- [~AIRINV\\_Service](#) ()
- void [buildSampleBom](#) ()
- void [clonePersistentBom](#) ()
- void [buildComplementaryLinks](#) (stdair::BomRoot &)
- stdair::RMEventList\_T [initRMEvents](#) (const stdair::Date\_T &iStartDate, const stdair::Date\_T &iEndDate)
- void [calculateAvailability](#) (stdair::TravelSolutionStruct &)
- bool [sell](#) (const std::string &iSegmentDateKey, const stdair::ClassCode\_T &, const stdair::PartySize\_T &)
- bool [sell](#) (const stdair::BookingClassID\_T &, const stdair::PartySize\_T &)
- bool [cancel](#) (const std::string &iSegmentDateKey, const stdair::ClassCode\_T &, const stdair::PartySize\_T &)
- bool [cancel](#) (const stdair::BookingClassID\_T &, const stdair::PartySize\_T &)
- void [takeSnapshots](#) (const stdair::AirlineCode\_T &, const stdair::DateTime\_T &)
- void [optimise](#) (const stdair::AirlineCode\_T &, const stdair::KeyDescription\_T &, const stdair::DateTime\_T &)
- std::string [jsonHandler](#) (const stdair::JSONString &) const
- std::string [jsonExportFlightDateList](#) (const stdair::AirlineCode\_T &iAirlineCode="all", const stdair::FlightNumber\_T &iFlightNumber=0) const
- std::string [jsonExportFlightDateObjects](#) (const stdair::AirlineCode\_T &, const stdair::FlightNumber\_T &, const stdair::Date\_T &iDepartureDate) const
- std::string [list](#) (const stdair::AirlineCode\_T &iAirlineCode="all", const stdair::FlightNumber\_T &iFlightNumber=0) const
- bool [check](#) (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 [AIRINV](#) Services.

Definition at line 40 of file AIRINV\_Service.hpp.

### 10.3.2 Constructor & Destructor Documentation

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

Constructor.

The `initAirinvService()` 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 89 of file AIRINV\_Service.cpp.

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

Constructor.

The initAirinvService() 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 60 of file AIRINV\_Service.cpp.

### 10.3.2.3 AIRINV::AIRINV\_Service::AIRINV\_Service (stdair::STDAIR\_ServicePtr\_T)

Constructor.

The initAirinvService() 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 [AIRINV\\_Master\\_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 120 of file AIRINV\_Service.cpp.

### 10.3.2.4 AIRINV::AIRINV\_Service::AIRINV\_Service (stdair::STDAIR\_ServicePtr\_T, SEVMGR::SEVMGR\_ServicePtr\_T)

Constructor.

The initAirinvService() 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 [AIRINV\\_Master\\_Service](#) is itself being initialised by another library service such as SIMCRS\_Service).

#### Parameters:

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

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

Definition at line 148 of file AIRINV\_Service.cpp.

### 10.3.2.5 AIRINV::AIRINV\_Service::~~AIRINV\_Service ()

Destructor.

Definition at line 177 of file AIRINV\_Service.cpp.

### 10.3.3 Member Function Documentation

#### 10.3.3.1 void AIRINV::AIRINV\_Service::parseAndLoad (const AIRINV::InventoryFilePath &)

Parse the inventory dump and load it into memory.

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

##### Parameters:

*const* AIRINV::InventoryFilePath& Filename of the input inventory file.

Definition at line 346 of file AIRINV\_Service.cpp.

References buildComplementaryLinks(), AIRINV::InventoryParser::buildInventory(), clonePersistentBom(), AIRINV::AIRINV\_ServiceContext::getOwnStdairServiceFlag(), and AIRINV::AIRINV\_ServiceContext::getSTDAIR\_Service().

Referenced by AIRINV::AIRINV\_Master\_Service::parseAndLoad().

#### 10.3.3.2 void AIRINV::AIRINV\_Service::parseAndLoad (const stdair::ScheduleFilePath &, const stdair::ODFilePath &, const stdair::FRAT5FilePath &, const stdair::FFDisutilityFilePath &, const AIRRAC::YieldFilePath &)

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.

*const* stdair::FRAT5FilePath& Filename of the input FRAT5 file.

*const* stdair::FFDisutilityFilePath& Filename of the input FF disutility file.

*const* AIRRAC::YieldFilePath& Filename of the input yield file.

Definition at line 404 of file AIRINV\_Service.cpp.

References buildComplementaryLinks(), clonePersistentBom(), AIRINV::ScheduleParser::generateInventories(), AIRINV::AIRINV\_ServiceContext::getAIRRAC\_Service(), AIRINV::AIRINV\_ServiceContext::getOwnStdairServiceFlag(), AIRINV::AIRINV\_ServiceContext::getSTDAIR\_Service(), AIRINV::FFDisutilityParser::parse(), and AIRINV::FRAT5Parser::parse().

#### 10.3.3.3 void AIRINV::AIRINV\_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 487 of file AIRINV\_Service.cpp.

References buildComplementaryLinks(), clonePersistentBom(), AIRINV::AIRINV\_ServiceContext::getAIRRAC\_Service(), AIRINV::AIRINV\_ServiceContext::getOwnStdairServiceFlag(), AIRINV::AIRINV\_ServiceContext::getRMOL\_Service(), and AIRINV::AIRINV\_ServiceContext::getSTDAIR\_Service().

Referenced by AIRINV::AIRINV\_Master\_Service::buildSampleBom().

**10.3.3.4 void AIRINV::AIRINV\_Service::clonePersistentBom ()**

Clone the persistent BOM object.

Definition at line 570 of file AIRINV\_Service.cpp.

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

Referenced by buildSampleBom(), AIRINV::AIRINV\_Master\_Service::clonePersistentBom(), and parseAndLoad().

**10.3.3.5 void AIRINV::AIRINV\_Service::buildComplementaryLinks (stdair::BomRoot &)**

Build all the complementary links in the given bom root object. For instance, build the links between leg and segment date (as well as leg and segment cabin).

Definition at line 608 of file AIRINV\_Service.cpp.

References AIRINV::InventoryManager::buildSimilarSegmentCabinSets(), AIRINV::InventoryManager::createDirectAccesses(), AIRINV::InventoryManager::initialiseListsOfUsablePolicies(), AIRINV::InventoryManager::initialiseYieldBasedNestingStructures(), and AIRINV::InventoryManager::setDefaultBidPriceVector().

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

**10.3.3.6 stdair::RMEventList\_T AIRINV::AIRINV\_Service::initRMEvents (const stdair::Date\_T & iStartDate, const stdair::Date\_T & iEndDate)**

Initialise the RM events for the inventory.

**Parameters:**

*const* stdair::Date\_T& Parameters for the start date.

*const* stdair::Date\_T& Parameters for the end date.

Definition at line 920 of file AIRINV\_Service.cpp.

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

Referenced by AIRINV::AIRINV\_Master\_Service::initSnapshotAndRMEvents().

**10.3.3.7 void AIRINV::AIRINV\_Service::calculateAvailability (stdair::TravelSolutionStruct &)**

Compute the availability for the given travel solution.

Definition at line 952 of file AIRINV\_Service.cpp.

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

Referenced by AIRINV::AIRINV\_Master\_Service::calculateAvailability().

**10.3.3.8 bool AIRINV::AIRINV\_Service::sell (const std::string & iSegmentDateKey, const stdair::ClassCode\_T &, const stdair::PartySize\_T &)**

Register a booking.

**Parameters:**

*const* std::string& Key for the segment on which the sale is made

**const** stdair::ClassCode\_T& Class code where the sale is made

**const** stdair::PartySize\_T& Party size

**Returns:**

bool Whether or not the sale was successfull

Definition at line 978 of file AIRINV\_Service.cpp.

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

Referenced by AIRINV::AIRINV\_Master\_Service::sell().

### 10.3.3.9 bool AIRINV::AIRINV\_Service::sell (const stdair::BookingClassID\_T &, const stdair::PartySize\_T &)

Register a booking.

**Parameters:**

**const** stdair::BookingClassID\_T&

**const** stdair::PartySize\_T& Party size

**Returns:**

bool Whether or not the sale was successfull

Definition at line 1019 of file AIRINV\_Service.cpp.

### 10.3.3.10 bool AIRINV::AIRINV\_Service::cancel (const std::string & iSegmentDateKey, const stdair::ClassCode\_T &, const stdair::PartySize\_T &)

Register a cancellation.

**Parameters:**

**const** std::string& Key for the segment on which the cancellation is made

**const** stdair::ClassCode\_T& Class code where the sale is made

**const** stdair::PartySize\_T& Party size

**Returns:**

bool Whether or not the cancellation was successfull

Definition at line 1036 of file AIRINV\_Service.cpp.

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

Referenced by AIRINV::AIRINV\_Master\_Service::cancel().

### 10.3.3.11 bool AIRINV::AIRINV\_Service::cancel (const stdair::BookingClassID\_T &, const stdair::PartySize\_T &)

Register a cancellation.

**Parameters:**

**const** stdair::BookingClassID\_T&  
**const** stdair::PartySize\_T& Party size

**Returns:**

bool Whether or not the cancellation was successful

Definition at line 1080 of file AIRINV\_Service.cpp.

**10.3.3.12 void AIRINV::AIRINV\_Service::takeSnapshots (const stdair::AirlineCode\_T &, const stdair::DateTime\_T &)**

Take inventory snapshots.

Definition at line 1097 of file AIRINV\_Service.cpp.

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

Referenced by AIRINV::AIRINV\_Master\_Service::takeSnapshots().

**10.3.3.13 void AIRINV::AIRINV\_Service::optimise (const stdair::AirlineCode\_T &, const stdair::KeyDescription\_T &, const stdair::DateTime\_T &)**

Optimise (revenue management) an flight-date/network-date

Definition at line 1124 of file AIRINV\_Service.cpp.

References AIRINV::AIRINV\_ServiceContext::getRMOL\_Service(), and AIRINV::AIRINV\_ServiceContext::getSTDAIR\_Service().

Referenced by AIRINV::AIRINV\_Master\_Service::optimise().

**10.3.3.14 std::string AIRINV::AIRINV\_Service::jsonHandler (const stdair::JSONString &) const**

Dispatch the JSon command string to the right JSon Service, according to the JSon command type.

**Parameters:**

**const** stdair::JSONString& Input string which contained the JSon command string.

**Returns:**

std::string Output string in which the asking objects are logged/dumped in a JSon format.

Definition at line 646 of file AIRINV\_Service.cpp.

References csvDisplay(), jsonExportFlightDateList(), jsonExportFlightDateObjects(), and list().

Referenced by AIRINV::AIRINV\_Master\_Service::jsonHandler().

**10.3.3.15 std::string AIRINV::AIRINV\_Service::jsonExportFlightDateList (const stdair::AirlineCode\_T & iAirlineCode = "all", const stdair::FlightNumber\_T & iFlightNumber = 0) const**

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



**Parameters:**

**const** AirlineCode& Airline for which the flight-dates should be displayed. If set to "all" (default), all the inventories will be displayed.

**const** FlightNumber\_T& Flight number for which all the departure dates should be displayed. If set to 0 (the default), all the flight numbers will be displayed.

Definition at line 778 of file AIRINV\_Service.cpp.

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

Referenced by AIRINV::AIRINV\_Master\_Service::jsonExportFlightDateList(), and jsonHandler().

### 10.3.3.16 std::string AIRINV::AIRINV\_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-date 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 the flight to dump.

**Returns:**

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

Definition at line 801 of file AIRINV\_Service.cpp.

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

Referenced by AIRINV::AIRINV\_Master\_Service::jsonExportFlightDateObjects(), and jsonHandler().

### 10.3.3.17 std::string AIRINV::AIRINV\_Service::list (const stdair::AirlineCode\_T & iAirlineCode = "all", const stdair::FlightNumber\_T & iFlightNumber = 0) const

Display the list of flight-dates (contained within the BOM tree) corresponding to the parameters given as input.

**Parameters:**

**const** AirlineCode& Airline for which the flight-dates should be displayed. If set to "all" (the default), all the inventories will be displayed.

**const** FlightNumber\_T& Flight number for which all the departure dates should be displayed. If set to 0 (the default), all the flight numbers will be displayed.

**Returns:**

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

Definition at line 826 of file AIRINV\_Service.cpp.

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

Referenced by jsonHandler(), and AIRINV::AIRINV\_Master\_Service::list().

**10.3.3.18** `bool AIRINV::AIRINV_Service::check (const stdair::AirlineCode_T &, const stdair::FlightNumber_T &, const stdair::Date_T & iDepartureDate) const`

Check whether the given flight-date is a valid one.

**Parameters:**

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

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

*const* stdair::Date\_T& Departure date of the flight to check.

**Returns:**

bool Whether or not the given flight date is valid.

Definition at line 850 of file AIRINV\_Service.cpp.

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

Referenced by AIRINV::AIRINV\_Master\_Service::check().

**10.3.3.19** `std::string AIRINV::AIRINV_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 874 of file AIRINV\_Service.cpp.

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

Referenced by AIRINV::AIRINV\_Master\_Service::csvDisplay(), and jsonHandler().

**10.3.3.20** `std::string AIRINV::AIRINV_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-date corresponding to the parameters given as input.

**Parameters:**

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

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

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

**Returns:**

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

Definition at line 896 of file AIRINV\_Service.cpp.

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

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

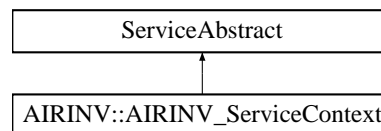
- [airinv/AIRINV\\_Service.hpp](#)
- [airinv/service/AIRINV\\_Service.cpp](#)

## 10.4 AIRINV::AIRINV\_ServiceContext Class Reference

Class holding the context of the AirInv services.

```
#include <airinv/service/AIRINV_ServiceContext.hpp>
```

Inheritance diagram for AIRINV::AIRINV\_ServiceContext::



### Friends

- class [AIRINV\\_Service](#)
- class [FacAirinvServiceContext](#)

### 10.4.1 Detailed Description

Class holding the context of the AirInv services.

Definition at line 28 of file AIRINV\_ServiceContext.hpp.

### 10.4.2 Friends And Related Function Documentation

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

The [AIRINV\\_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 34 of file AIRINV\_ServiceContext.hpp.

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

Definition at line 35 of file AIRINV\_ServiceContext.hpp.

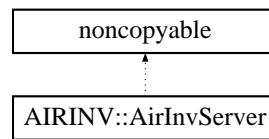
The documentation for this class was generated from the following files:

- airinv/service/[AIRINV\\_ServiceContext.hpp](#)
- airinv/service/[AIRINV\\_ServiceContext.cpp](#)

## 10.5 AIRINV::AirInvServer Class Reference

```
#include <airinv/server/AirInvServer.hpp>
```

Inheritance diagram for AIRINV::AirInvServer::



## Public Member Functions

- [AirInvServer](#) (const std::string &address, const std::string &port, const stdair::AirlineCode\_T &i-AirlineCode, std::size\_t thread\_pool\_size)
- [~AirInvServer](#) ()
- void [run](#) ()
- void [stop](#) ()

### 10.5.1 Detailed Description

The top-level class of the AirInv server.

Definition at line 23 of file AirInvServer.hpp.

### 10.5.2 Constructor & Destructor Documentation

#### 10.5.2.1 AIRINV::AirInvServer::AirInvServer (const std::string & address, const std::string & port, const stdair::AirlineCode\_T & iAirlineCode, std::size\_t thread\_pool\_size)

Constructor.

Construct the server to listen on the specified TCP address and port, and serve up files from the given directory.

Definition at line 20 of file AirInvServer\_ASIO.cpp.

#### 10.5.2.2 AIRINV::AirInvServer::~~AirInvServer ()

Destructor.

Definition at line 46 of file AirInvServer\_ASIO.cpp.

### 10.5.3 Member Function Documentation

#### 10.5.3.1 void AIRINV::AirInvServer::run ()

Run the server's io\_service loop.

Definition at line 50 of file AirInvServer\_ASIO.cpp.

Referenced by main().

#### 10.5.3.2 void AIRINV::AirInvServer::stop ()

Stop the server.

Definition at line 69 of file AirInvServer\_ASIO.cpp.

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

- [airinv/server/AirInvServer.hpp](#)
- [airinv/server/AirInvServer\\_ASIO.cpp](#)

## 10.6 AIRINV::BomAbstract Class Reference

```
#include <airinv/bom/BomAbstract.hpp>
```

### Public Member Functions

- virtual void [toStream](#) (std::ostream &ioOut) const =0
- virtual void [fromStream](#) (std::istream &ioIn)=0
- virtual std::string [toString](#) () const =0
- virtual std::string [describeKey](#) () const =0
- virtual std::string [describeShortKey](#) () const =0

### Protected Member Functions

- [BomAbstract](#) ()
- [BomAbstract](#) (const [BomAbstract](#) &)
- virtual [~BomAbstract](#) ()

### Friends

- class [FacBomAbstract](#)

#### 10.6.1 Detailed Description

Base class for the Business Object Model (BOM) layer.

Definition at line 14 of file BomAbstract.hpp.

#### 10.6.2 Constructor & Destructor Documentation

##### 10.6.2.1 AIRINV::BomAbstract::BomAbstract () [inline, protected]

Protected Default Constructor to ensure this class is abstract.

Definition at line 40 of file BomAbstract.hpp.

##### 10.6.2.2 AIRINV::BomAbstract::BomAbstract (const [BomAbstract](#) &) [inline, protected]

Definition at line 41 of file BomAbstract.hpp.

##### 10.6.2.3 virtual [AIRINV::BomAbstract::~~BomAbstract](#) () [inline, protected, virtual]

Destructor.

Definition at line 44 of file BomAbstract.hpp.

### 10.6.3 Member Function Documentation

**10.6.3.1** `virtual void AIRINV::BomAbstract::toStream (std::ostream & ioOut) const` [pure virtual]

Dump a Business Object into an output stream.

**Parameters:**

*ostream&* the output stream.

**10.6.3.2** `virtual void AIRINV::BomAbstract::fromStream (std::istream & ioIn)` [pure virtual]

Read a Business Object from an input stream.

**Parameters:**

*istream&* the input stream.

Referenced by operator>>().

**10.6.3.3** `virtual std::string AIRINV::BomAbstract::toString () const` [pure virtual]

Get the serialised version of the Business Object.

**10.6.3.4** `virtual std::string AIRINV::BomAbstract::describeKey () const` [pure virtual]

Get a string describing the whole key (differentiating two objects at any level).

**10.6.3.5** `virtual std::string AIRINV::BomAbstract::describeShortKey () const` [pure virtual]

Get a string describing the short key (differentiating two objects at the same level).

### 10.6.4 Friends And Related Function Documentation

**10.6.4.1** `friend class FacBomAbstract` [friend]

Definition at line 15 of file BomAbstract.hpp.

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

- [airinv/bom/BomAbstract.hpp](#)

## 10.7 stdair::BomPropertyTree Struct Reference

```
#include <airinv/server/BomPropertyTree.hpp>
```

### Public Member Functions

- void [load](#) (const std::string &iBomTree)
- std::string [save](#) () const

## Public Attributes

- `stdair::AirlineCode_T` [\\_airlineCode](#)
- `stdair::FlightNumber_T` [\\_flightNumber](#)
- `stdair::Date_T` [\\_departureDate](#)
- `std::set< stdair::AirportCode_T >` [\\_airportCodeList](#)

### 10.7.1 Detailed Description

Structure representing a list of airports.

Definition at line 19 of file BomPropertyTree.hpp.

### 10.7.2 Member Function Documentation

#### 10.7.2.1 `void stdair::BomPropertyTree::load (const std::string & iBomTree)`

Update the current BOM tree (\*this) with the parsed stream, which is JSON formatted.

Definition at line 17 of file BomPropertyTree.cpp.

References [\\_airlineCode](#), [\\_departureDate](#), and [\\_flightNumber](#).

#### 10.7.2.2 `std::string stdair::BomPropertyTree::save () const`

Dump the BOM tree (\*this) into the stream with a JSON format.

Definition at line 60 of file BomPropertyTree.cpp.

References [\\_airlineCode](#), [\\_airportCodeList](#), [\\_departureDate](#), and [\\_flightNumber](#).

### 10.7.3 Member Data Documentation

#### 10.7.3.1 `stdair::AirlineCode_T` [stdair::BomPropertyTree::\\_airlineCode](#)

Airline code.

Definition at line 33 of file BomPropertyTree.hpp.

Referenced by [load\(\)](#), and [save\(\)](#).

#### 10.7.3.2 `stdair::FlightNumber_T` [stdair::BomPropertyTree::\\_flightNumber](#)

Flight number.

Definition at line 36 of file BomPropertyTree.hpp.

Referenced by [load\(\)](#), and [save\(\)](#).

#### 10.7.3.3 `stdair::Date_T` [stdair::BomPropertyTree::\\_departureDate](#)

Departure date.

Definition at line 39 of file BomPropertyTree.hpp.

Referenced by [load\(\)](#), and [save\(\)](#).

#### 10.7.3.4 std::set<stdair::AirportCode\_T> stdair::BomPropertyTree::\_airportCodeList

Just to have a list, for now.

Definition at line 42 of file BomPropertyTree.hpp.

Referenced by save().

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

- [airinv/server/BomPropertyTree.hpp](#)
- [airinv/server/BomPropertyTree.cpp](#)

## 10.8 AIRINV::BomRootHelper Class Reference

```
#include <airinv/bom/BomRootHelper.hpp>
```

### Static Public Member Functions

- static void [fillFromRouting](#) (const stdair::BomRoot &)

#### 10.8.1 Detailed Description

Class representing the actual business functions for an airline bom root.

Definition at line 16 of file BomRootHelper.hpp.

#### 10.8.2 Member Function Documentation

##### 10.8.2.1 void AIRINV::BomRootHelper::fillFromRouting (const stdair::BomRoot &) [static]

Fill the attributes derived from the routing legs (e.g., board and off dates).

Definition at line 16 of file BomRootHelper.cpp.

References AIRINV::InventoryHelper::fillFromRouting().

Referenced by AIRINV::InventoryManager::createDirectAccesses().

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

- [airinv/bom/BomRootHelper.hpp](#)
- [airinv/bom/BomRootHelper.cpp](#)

## 10.9 AIRINV::BookingClassHelper Class Reference

```
#include <airinv/bom/BookingClassHelper.hpp>
```

#### 10.9.1 Detailed Description

Class representing the actual business functions for an airline booking class.

Definition at line 19 of file BookingClassHelper.hpp.

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

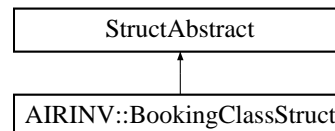


- [airinv/bom/BookingClassHelper.hpp](#)

## 10.10 AIRINV::BookingClassStruct Struct Reference

```
#include <airinv/bom/BookingClassStruct.hpp>
```

Inheritance diagram for AIRINV::BookingClassStruct::



### Public Member Functions

- stdair::ClassCode\_T [getFullSubclassCode](#) () const
- void [fill](#) (stdair::BookingClass &) const
- const std::string [describe](#) () const
- [BookingClassStruct](#) ()

### Public Attributes

- stdair::ClassCode\_T [\\_classCode](#)
- stdair::SubclassCode\_T [\\_subclassCode](#)
- stdair::ClassCode\_T [\\_parentClassCode](#)
- stdair::SubclassCode\_T [\\_parentSubclassCode](#)
- stdair::AuthorizationLevel\_T [\\_cumulatedProtection](#)
- stdair::AuthorizationLevel\_T [\\_protection](#)
- stdair::NbOfSeats\_T [\\_nego](#)
- stdair::OverbookingRate\_T [\\_noShowPercentage](#)
- stdair::OverbookingRate\_T [\\_overbookingPercentage](#)
- stdair::NbOfBookings\_T [\\_nbOfBookings](#)
- stdair::NbOfBookings\_T [\\_nbOfGroupBookings](#)
- stdair::NbOfBookings\_T [\\_nbOfPendingGroupBookings](#)
- stdair::NbOfBookings\_T [\\_nbOfStaffBookings](#)
- stdair::NbOfBookings\_T [\\_nbOfWLBookings](#)
- stdair::NbOfBookings\_T [\\_etb](#)
- stdair::Availability\_T [\\_netClassAvailability](#)
- stdair::Availability\_T [\\_segmentAvailability](#)
- stdair::Availability\_T [\\_netRevenueAvailability](#)

#### 10.10.1 Detailed Description

Utility Structure for the parsing of BookingClass structures.

Definition at line 24 of file BookingClassStruct.hpp.

## 10.10.2 Constructor & Destructor Documentation

### 10.10.2.1 AIRINV::BookingClassStruct::BookingClassStruct ()

Default Constructor.

Definition at line 16 of file BookingClassStruct.cpp.

## 10.10.3 Member Function Documentation

### 10.10.3.1 stdair::ClassCode\_T AIRINV::BookingClassStruct::getFullSubclassCode () const

Returns the concatenation of the class and subclass codes.

Definition at line 20 of file BookingClassStruct.cpp.

References `_classCode`, and `_subclassCode`.

### 10.10.3.2 void AIRINV::BookingClassStruct::fill (stdair::BookingClass &) const

Fill the BookingClass objects with the attributes of the [BookingClassStruct](#).

Definition at line 44 of file BookingClassStruct.cpp.

### 10.10.3.3 const std::string AIRINV::BookingClassStruct::describe () const

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

Definition at line 27 of file BookingClassStruct.cpp.

References `_classCode`, `_cumulatedProtection`, `_etb`, `_nbOfBookings`, `_nbOfGroupBookings`, `_nbOfPendingGroupBookings`, `_nbOfStaffBookings`, `_nbOfWLBookings`, `_nego`, `_netClassAvailability`, `_netRevenueAvailability`, `_noShowPercentage`, `_overbookingPercentage`, `_parentClassCode`, `_parentSubclassCode`, `_protection`, `_segmentAvailability`, and `_subclassCode`.

## 10.10.4 Member Data Documentation

### 10.10.4.1 stdair::ClassCode\_T AIRINV::BookingClassStruct::\_classCode

Definition at line 26 of file BookingClassStruct.hpp.

Referenced by `describe()`, `getFullSubclassCode()`, `AIRINV::InventoryParserHelper::storeFCClasses::operator()`, `AIRINV::InventoryParserHelper::storeClassCode::operator()`, `AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator()`, `AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator()`, and `AIRINV::InventoryParserHelper::storeAirlineCode::operator()`.

### 10.10.4.2 stdair::SubclassCode\_T AIRINV::BookingClassStruct::\_subclassCode

Definition at line 27 of file BookingClassStruct.hpp.

Referenced by `describe()`, `getFullSubclassCode()`, and `AIRINV::InventoryParserHelper::storeSubclassCode::operator()`.

### 10.10.4.3 stdair::ClassCode\_T AIRINV::BookingClassStruct::\_parentClassCode

Definition at line 28 of file BookingClassStruct.hpp.

Referenced by describe(), and AIRINV::InventoryParserHelper::storeParentClassCode::operator()().

#### 10.10.4.4 stdair::SubclassCode\_T [AIRINV::BookingClassStruct::\\_parentSubclassCode](#)

Definition at line 29 of file BookingClassStruct.hpp.

Referenced by describe(), and AIRINV::InventoryParserHelper::storeParentSubclassCode::operator()().

#### 10.10.4.5 stdair::AuthorizationLevel\_T [AIRINV::BookingClassStruct::\\_cumulatedProtection](#)

Definition at line 30 of file BookingClassStruct.hpp.

Referenced by describe(), and AIRINV::InventoryParserHelper::storeCumulatedProtection::operator()().

#### 10.10.4.6 stdair::AuthorizationLevel\_T [AIRINV::BookingClassStruct::\\_protection](#)

Definition at line 31 of file BookingClassStruct.hpp.

Referenced by describe(), and AIRINV::InventoryParserHelper::storeProtection::operator()().

#### 10.10.4.7 stdair::NbOfSeats\_T [AIRINV::BookingClassStruct::\\_nego](#)

Definition at line 32 of file BookingClassStruct.hpp.

Referenced by describe(), and AIRINV::InventoryParserHelper::storeNego::operator()().

#### 10.10.4.8 stdair::OverbookingRate\_T [AIRINV::BookingClassStruct::\\_noShowPercentage](#)

Definition at line 33 of file BookingClassStruct.hpp.

Referenced by describe(), and AIRINV::InventoryParserHelper::storeNoShow::operator()().

#### 10.10.4.9 stdair::OverbookingRate\_T [AIRINV::BookingClassStruct::\\_overbookingPercentage](#)

Definition at line 34 of file BookingClassStruct.hpp.

Referenced by describe(), and AIRINV::InventoryParserHelper::storeOverbooking::operator()().

#### 10.10.4.10 stdair::NbOfBookings\_T [AIRINV::BookingClassStruct::\\_nbOfBookings](#)

Definition at line 35 of file BookingClassStruct.hpp.

Referenced by describe(), and AIRINV::InventoryParserHelper::storeNbOfBkgs::operator()().

#### 10.10.4.11 stdair::NbOfBookings\_T [AIRINV::BookingClassStruct::\\_nbOfGroupBookings](#)

Definition at line 36 of file BookingClassStruct.hpp.

Referenced by describe(), and AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator()().

#### 10.10.4.12 stdair::NbOfBookings\_T [AIRINV::BookingClassStruct::\\_nbOfPendingGroupBookings](#)

Definition at line 37 of file BookingClassStruct.hpp.

Referenced by describe(), and AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator()().

**10.10.4.13** `stdair::NbOfBookings_T` [AIRINV::BookingClassStruct::\\_nbOfStaffBookings](#)

Definition at line 38 of file `BookingClassStruct.hpp`.

Referenced by `describe()`, and `AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator()()`.

**10.10.4.14** `stdair::NbOfBookings_T` [AIRINV::BookingClassStruct::\\_nbOfWLBkgs](#)

Definition at line 39 of file `BookingClassStruct.hpp`.

Referenced by `describe()`, and `AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator()()`.

**10.10.4.15** `stdair::NbOfBookings_T` [AIRINV::BookingClassStruct::\\_etb](#)

Definition at line 40 of file `BookingClassStruct.hpp`.

Referenced by `describe()`, and `AIRINV::InventoryParserHelper::storeClassETB::operator()()`.

**10.10.4.16** `stdair::Availability_T` [AIRINV::BookingClassStruct::\\_netClassAvailability](#)

Definition at line 41 of file `BookingClassStruct.hpp`.

Referenced by `describe()`, and `AIRINV::InventoryParserHelper::storeClassAvailability::operator()()`.

**10.10.4.17** `stdair::Availability_T` [AIRINV::BookingClassStruct::\\_segmentAvailability](#)

Definition at line 42 of file `BookingClassStruct.hpp`.

Referenced by `describe()`, and `AIRINV::InventoryParserHelper::storeSegmentAvailability::operator()()`.

**10.10.4.18** `stdair::Availability_T` [AIRINV::BookingClassStruct::\\_netRevenueAvailability](#)

Definition at line 43 of file `BookingClassStruct.hpp`.

Referenced by `describe()`, and `AIRINV::InventoryParserHelper::storeRevenueAvailability::operator()()`.

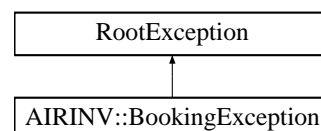
The documentation for this struct was generated from the following files:

- `airinv/bom/BookingClassStruct.hpp`
- `airinv/bom/BookingClassStruct.cpp`

**10.11** **AIRINV::BookingException Class Reference**

```
#include <airinv/AIRINV_Types.hpp>
```

Inheritance diagram for `AIRINV::BookingException`:



### 10.11.1 Detailed Description

Specific exception related to bookings made against the inventory.

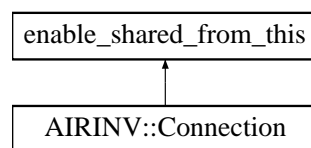
Definition at line 166 of file AIRINV\_Types.hpp.

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

- [airinv/AIRINV\\_Types.hpp](#)

## 10.12 enable\_shared\_from\_this Class Reference

Inheritance diagram for enable\_shared\_from\_this::

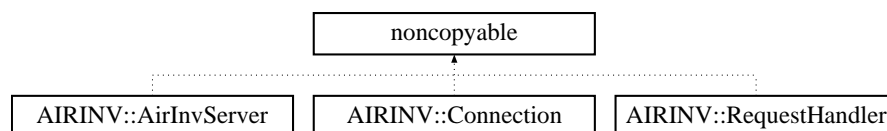


The documentation for this class was generated from the following file:

- [airinv/server/Connection.hpp](#)

## 10.13 noncopyable Class Reference

Inheritance diagram for noncopyable::

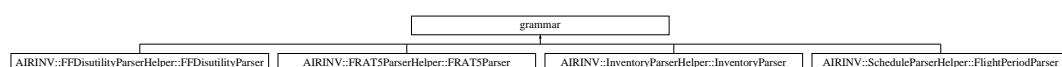


The documentation for this class was generated from the following files:

- [airinv/server/Connection.hpp](#)
- [airinv/server/AirInvServer.hpp](#)
- [airinv/server/RequestHandler.hpp](#)

## 10.14 grammar Class Reference

Inheritance diagram for grammar::

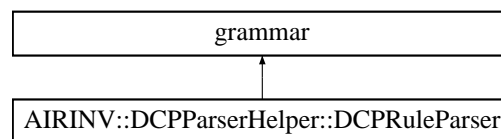


The documentation for this class was generated from the following files:

- [airinv/command/FFDisutilityParserHelper.hpp](#)
- [airinv/command/ScheduleParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/FRAT5ParserHelper.hpp](#)

## 10.15 grammar Class Reference

Inheritance diagram for grammar::



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

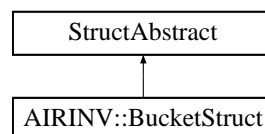
- [airinv/command/vault/DCPParserHelper.hpp](#)

## 10.16 AIRINV::BucketStruct Struct Reference

Utility Structure for the parsing of Bucket structures.

```
#include <airinv/bom/BucketStruct.hpp>
```

Inheritance diagram for AIRINV::BucketStruct::



### Public Member Functions

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

### Public Attributes

- stdair::Yield\_T [\\_yieldRangeUpperValue](#)
- stdair::CabinCapacity\_T [\\_availability](#)
- stdair::NbOfSeats\_T [\\_nbOfSeats](#)
- stdair::SeatIndex\_T [\\_seatIndex](#)

### 10.16.1 Detailed Description

Utility Structure for the parsing of Bucket structures.

Definition at line 26 of file BucketStruct.hpp.

### 10.16.2 Constructor & Destructor Documentation

#### 10.16.2.1 AIRINV::BucketStruct::BucketStruct ()

Default Constructor.

Definition at line 16 of file BucketStruct.cpp.

### 10.16.3 Member Function Documentation

#### 10.16.3.1 void AIRINV::BucketStruct::fill (stdair::Bucket &) const

Fill the Bucket objects with the attributes of the [BucketStruct](#).

Definition at line 29 of file BucketStruct.cpp.

References `_availability`, `_nbOfSeats`, and `_yieldRangeUpperValue`.

#### 10.16.3.2 const std::string AIRINV::BucketStruct::describe () const

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

Definition at line 20 of file BucketStruct.cpp.

References `_availability`, `_nbOfSeats`, `_seatIndex`, and `_yieldRangeUpperValue`.

### 10.16.4 Member Data Documentation

#### 10.16.4.1 stdair::Yield\_T AIRINV::BucketStruct::\_yieldRangeUpperValue

Definition at line 28 of file BucketStruct.hpp.

Referenced by `describe()`, `fill()`, `AIRINV::InventoryParserHelper::storeYieldUpperRange::operator()`, `AIRINV::InventoryParserHelper::storeLegCabinCode::operator()`, `AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator()`, and `AIRINV::InventoryParserHelper::storeAirlineCode::operator()`.

#### 10.16.4.2 stdair::CabinCapacity\_T AIRINV::BucketStruct::\_availability

Definition at line 29 of file BucketStruct.hpp.

Referenced by `describe()`, `fill()`, and `AIRINV::InventoryParserHelper::storeBucketAvailability::operator()`.

#### 10.16.4.3 stdair::NbOfSeats\_T AIRINV::BucketStruct::\_nbOfSeats

Definition at line 30 of file BucketStruct.hpp.

Referenced by `describe()`, and `fill()`.

#### 10.16.4.4 stdair::SeatIndex\_T AIRINV::BucketStruct::\_seatIndex

Definition at line 31 of file BucketStruct.hpp.

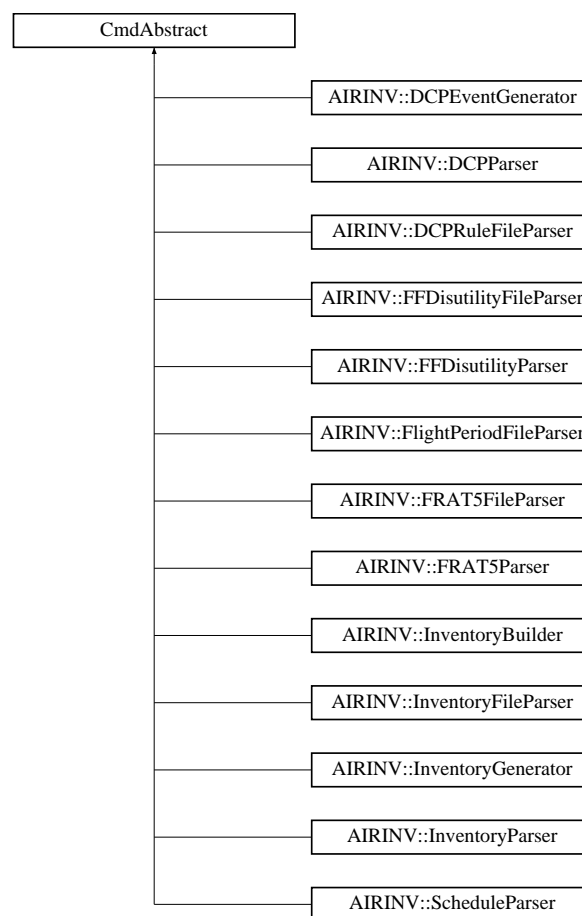
Referenced by describe(), and AIRINV::InventoryParserHelper::storeSeatIndex::operator().

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

- [airinv/bom/BucketStruct.hpp](#)
- [airinv/bom/BucketStruct.cpp](#)

## 10.17 CmdAbstract Class Reference

Inheritance diagram for CmdAbstract::



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

- [airinv/command/FFDisutilityParser.hpp](#)
- [airinv/command/InventoryBuilder.hpp](#)
- [airinv/command/InventoryParser.hpp](#)
- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/ScheduleParserHelper.hpp](#)
- [airinv/command/ScheduleParser.hpp](#)

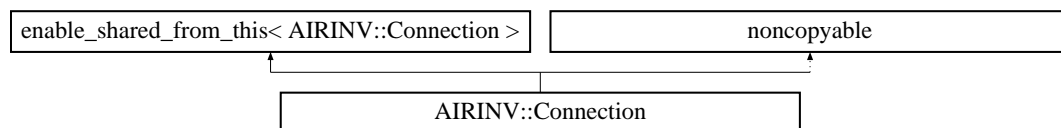


- [airinv/command/vault/DCPParser.hpp](#)
- [airinv/command/FRAT5Parser.hpp](#)
- [airinv/command/FFDisutilityParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/vault/DCPEventGenerator.hpp](#)
- [airinv/command/FRAT5ParserHelper.hpp](#)
- [airinv/command/InventoryGenerator.hpp](#)

## 10.18 AIRINV::Connection Class Reference

```
#include <airinv/server/Connection.hpp>
```

Inheritance diagram for AIRINV::Connection::



### Public Member Functions

- [Connection](#) (boost::asio::io\_service &, [RequestHandler](#) &)
- boost::asio::ip::tcp::socket & [socket](#) ()
- void [start](#) ()

### 10.18.1 Detailed Description

Represents a single connection from a client.

Definition at line 25 of file Connection.hpp.

### 10.18.2 Constructor & Destructor Documentation

#### 10.18.2.1 AIRINV::Connection::Connection (boost::asio::io\_service &, [RequestHandler](#) &)

Constructor.

Construct a connection with the given io\_service.

Definition at line 16 of file Connection.cpp.

### 10.18.3 Member Function Documentation

#### 10.18.3.1 boost::asio::ip::tcp::socket & AIRINV::Connection::socket ()

Get the socket associated with the connection.

Definition at line 22 of file Connection.cpp.

### 10.18.3.2 void AIRINV::Connection::start ()

Start the first asynchronous operation for the connection.

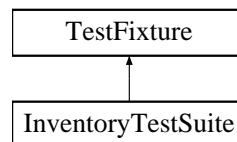
Definition at line 27 of file Connection.cpp.

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

- [airinv/server/Connection.hpp](#)
- [airinv/server/Connection.cpp](#)

## 10.19 TestFixture Class Reference

Inheritance diagram for TestFixture::



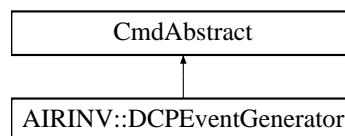
The documentation for this class was generated from the following file:

- [test/airinv/InventoryTestSuite.hpp](#)

## 10.20 AIRINV::DCPEventGenerator Class Reference

```
#include <airinv/command/vault/DCPEventGenerator.hpp>
```

Inheritance diagram for AIRINV::DCPEventGenerator::



### Friends

- class [DCPFileParser](#)
- struct [DCPParserHelper::doEndDCP](#)
- class [DCPParser](#)

### 10.20.1 Detailed Description

Class handling the generation / instantiation of the DCP BOM.

Definition at line 27 of file DCPEventGenerator.hpp.

## 10.20.2 Friends And Related Function Documentation

### 10.20.2.1 friend class DCPFileParser [friend]

Definition at line 31 of file DCPEventGenerator.hpp.

### 10.20.2.2 friend struct DCPParserHelper::doEndDCP [friend]

Definition at line 32 of file DCPEventGenerator.hpp.

### 10.20.2.3 friend class DCPParser [friend]

Definition at line 33 of file DCPEventGenerator.hpp.

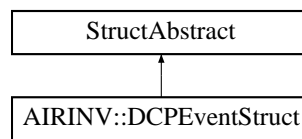
The documentation for this class was generated from the following files:

- [airinv/command/vault/DCPEventGenerator.hpp](#)
- [airinv/command/vault/DCPEventGenerator.cpp](#)

## 10.21 AIRINV::DCPEventStruct Struct Reference

```
#include <airinv/bom/DCPEventStruct.hpp>
```

Inheritance diagram for AIRINV::DCPEventStruct::



### Public Member Functions

- [DCPEventStruct \(\)](#)
- [stdair::Date\\_T getDate \(\) const](#)
- [stdair::Duration\\_T getTime \(\) const](#)
- [const std::string describe \(\) const](#)
- [const unsigned int getAirlineListSize \(\) const](#)
- [const unsigned int getClassCodeListSize \(\) const](#)
- [const stdair::AirlineCode\\_T & getFirstAirlineCode \(\) const](#)
- [void beginAirline \(\)](#)
- [bool hasNotReachedEndAirline \(\) const](#)
- [stdair::AirlineCode\\_T getCurrentAirlineCode \(\) const](#)
- [void iterateAirline \(\)](#)
- [const std::string & getFirstClassCode \(\) const](#)
- [void beginClassCode \(\)](#)
- [bool hasNotReachedEndClassCode \(\) const](#)
- [std::string getCurrentClassCode \(\) const](#)
- [void iterateClassCode \(\)](#)

## Public Attributes

- [stdair::year\\_t \\_itYear](#)
- [stdair::month\\_t \\_itMonth](#)
- [stdair::day\\_t \\_itDay](#)
- [stdair::hour\\_t \\_itHours](#)
- [stdair::minute\\_t \\_itMinutes](#)
- [stdair::second\\_t \\_itSeconds](#)
- [stdair::AirlineCodeList\\_T::iterator \\_itCurrentAirlineCode](#)
- [stdair::ClassList\\_StringList\\_T::iterator \\_itCurrentClassCode](#)
- [stdair::AirportCode\\_T \\_origin](#)
- [stdair::AirportCode\\_T \\_destination](#)
- [stdair::Date\\_T \\_dateRangeStart](#)
- [stdair::Date\\_T \\_dateRangeEnd](#)
- [stdair::Duration\\_T \\_timeRangeStart](#)
- [stdair::Duration\\_T \\_timeRangeEnd](#)
- [stdair::CabinCode\\_T \\_cabinCode](#)
- [stdair::CityCode\\_T \\_pos](#)
- [stdair::ChannelLabel\\_T \\_channel](#)
- [stdair::DayDuration\\_T \\_advancePurchase](#)
- [stdair::SaturdayStay\\_T \\_saturdayStay](#)
- [stdair::ChangeFees\\_T \\_changeFees](#)
- [stdair::NonRefundable\\_T \\_nonRefundable](#)
- [stdair::DayDuration\\_T \\_minimumStay](#)
- [stdair::PriceValue\\_T \\_DCP](#)
- [stdair::AirlineCode\\_T \\_airlineCode](#)
- [stdair::ClassCode\\_T \\_classCode](#)
- [stdair::AirlineCodeList\\_T \\_airlineCodeList](#)
- [stdair::ClassList\\_StringList\\_T \\_classCodeList](#)

### 10.21.1 Detailed Description

Utility Structure for the parsing of Flight-Period structures.

Definition at line 21 of file DCPEventStruct.hpp.

### 10.21.2 Constructor & Destructor Documentation

#### 10.21.2.1 AIRINV::DCPEventStruct::DCPEventStruct ()

Default constructor.

Definition at line 18 of file DCPEventStruct.cpp.

### 10.21.3 Member Function Documentation

#### 10.21.3.1 stdair::Date\_T AIRINV::DCPEventStruct::getDate () const

Get the date from the staging details.

Definition at line 38 of file DCPEventStruct.cpp.

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

**10.21.3.2 stdair::Duration\_T AIRINV::DCPEventStruct::getTime () const**

Get the time from the staging details.

Definition at line 44 of file DCPEventStruct.cpp.

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

**10.21.3.3 const std::string AIRINV::DCPEventStruct::describe () const**

Display of the structure.

Definition at line 53 of file DCPEventStruct.cpp.

References `_advancePurchase`, `_airlineCodeList`, `_cabinCode`, `_changeFees`, `_channel`, `_classCodeList`, `_dateRangeEnd`, `_dateRangeStart`, `_DCP`, `_destination`, `_minimumStay`, `_nonRefundable`, `_origin`, `_pos`, `_saturdayStay`, `_timeRangeEnd`, and `_timeRangeStart`.

**10.21.3.4 const unsigned int AIRINV::DCPEventStruct::getAirlineListSize () const [inline]**

Get the size of the airline code list.

Definition at line 37 of file DCPEventStruct.hpp.

References `_airlineCodeList`.

**10.21.3.5 const unsigned int AIRINV::DCPEventStruct::getClassCodeListSize () const [inline]**

Get the size of the class code list.

Definition at line 42 of file DCPEventStruct.hpp.

References `_classCodeList`.

**10.21.3.6 const stdair::AirlineCode\_T & AIRINV::DCPEventStruct::getFirstAirlineCode () const**

Get the first airline code.

Definition at line 87 of file DCPEventStruct.cpp.

References `_airlineCodeList`.

**10.21.3.7 void AIRINV::DCPEventStruct::beginAirline ()**

Initialise the internal iterators on airline code: The current iterator is set on the first airline code, the next iterator is set on the second one.

Definition at line 95 of file DCPEventStruct.cpp.

References `_airlineCodeList`, and `_itCurrentAirlineCode`.

**10.21.3.8 bool AIRINV::DCPEventStruct::hasNotReachedEndAirline () const**

States whether or not the end of the (airline code) list has been reached.

Definition at line 100 of file DCPEventStruct.cpp.

References `_airlineCodeList`, and `_itCurrentAirlineCode`.

**10.21.3.9 stdair::AirlineCode\_T AIRINV::DCPEventStruct::getCurrentAirlineCode () const**

Get the current element (airline code).

Definition at line 106 of file DCPEventStruct.cpp.

References `_airlineCodeList`, and `_itCurrentAirlineCode`.

**10.21.3.10 void AIRINV::DCPEventStruct::iterateAirline ()**

Iterate for one element (airline code): increment both internal iterators on Buckets.

Definition at line 112 of file DCPEventStruct.cpp.

References `_classCodeList`, and `_itCurrentAirlineCode`.

**10.21.3.11 const std::string & AIRINV::DCPEventStruct::getFirstClassCode () const**

Get the first class code list as a string.

Definition at line 119 of file DCPEventStruct.cpp.

References `_classCodeList`.

**10.21.3.12 void AIRINV::DCPEventStruct::beginClassCode ()**

Initialise the internal iterators on class code: The current iterator is set on the first class code, the next iterator is set on the second one.

Definition at line 127 of file DCPEventStruct.cpp.

References `_classCodeList`, and `_itCurrentClassCode`.

**10.21.3.13 bool AIRINV::DCPEventStruct::hasNotReachedEndClassCode () const**

States whether or not the end of the (class code) list has been reached.

Definition at line 132 of file DCPEventStruct.cpp.

References `_classCodeList`, and `_itCurrentClassCode`.

**10.21.3.14 std::string AIRINV::DCPEventStruct::getCurrentClassCode () const**

Get the current element (class code).

Definition at line 138 of file DCPEventStruct.cpp.

References `_classCodeList`, and `_itCurrentClassCode`.

**10.21.3.15 void AIRINV::DCPEventStruct::iterateClassCode ()**

Iterate for one element (classCode): increment both internal iterators on Buckets.

Definition at line 145 of file DCPEventStruct.cpp.

References `_classCodeList`, and `_itCurrentClassCode`.

**10.21.4 Member Data Documentation****10.21.4.1 stdair::year\_t AIRINV::DCPEventStruct::\_itYear**

Staging Date.

Definition at line 87 of file DCPEventStruct.hpp.

Referenced by getDate().

#### 10.21.4.2 stdair::month\_t AIRINV::DCPEventStruct::\_itMonth

Definition at line 88 of file DCPEventStruct.hpp.

Referenced by getDate().

#### 10.21.4.3 stdair::day\_t AIRINV::DCPEventStruct::\_itDay

Definition at line 89 of file DCPEventStruct.hpp.

Referenced by getDate().

#### 10.21.4.4 stdair::hour\_t AIRINV::DCPEventStruct::\_itHours

Staging Time.

Definition at line 93 of file DCPEventStruct.hpp.

Referenced by getTime().

#### 10.21.4.5 stdair::minute\_t AIRINV::DCPEventStruct::\_itMinutes

Definition at line 94 of file DCPEventStruct.hpp.

Referenced by getTime().

#### 10.21.4.6 stdair::second\_t AIRINV::DCPEventStruct::\_itSeconds

Definition at line 95 of file DCPEventStruct.hpp.

Referenced by getTime().

#### 10.21.4.7 stdair::AirlineCodeList\_T::iterator AIRINV::DCPEventStruct::\_itCurrentAirlineCode

Iterator for the current airline code list.

Definition at line 98 of file DCPEventStruct.hpp.

Referenced by beginAirline(), getCurrentAirlineCode(), hasNotReachedEndAirline(), and iterateAirline().

#### 10.21.4.8 stdair::ClassList\_StringList\_T::iterator AIRINV::DCPEventStruct::\_itCurrentClass-Code

Iterator for the current class code.

Definition at line 101 of file DCPEventStruct.hpp.

Referenced by beginClassCode(), getCurrentClassCode(), hasNotReachedEndClassCode(), and iterateClassCode().

**10.21.4.9 stdair::AirportCode\_T AIRINV::DCPEventStruct::\_origin**

Origin.

Definition at line 104 of file DCPEventStruct.hpp.

Referenced by describe().

**10.21.4.10 stdair::AirportCode\_T AIRINV::DCPEventStruct::\_destination**

Destination.

Definition at line 107 of file DCPEventStruct.hpp.

Referenced by describe().

**10.21.4.11 stdair::Date\_T AIRINV::DCPEventStruct::\_dateRangeStart**

Start Range date available for this DCP event.

Definition at line 110 of file DCPEventStruct.hpp.

Referenced by describe().

**10.21.4.12 stdair::Date\_T AIRINV::DCPEventStruct::\_dateRangeEnd**

Start Range date available for this DCP event.

Definition at line 113 of file DCPEventStruct.hpp.

Referenced by describe().

**10.21.4.13 stdair::Duration\_T AIRINV::DCPEventStruct::\_timeRangeStart**

Start time from the time range available for this DCP event.

Definition at line 116 of file DCPEventStruct.hpp.

Referenced by describe().

**10.21.4.14 stdair::Duration\_T AIRINV::DCPEventStruct::\_timeRangeEnd**

End time from the time range available for this DCP event.

Definition at line 119 of file DCPEventStruct.hpp.

Referenced by describe().

**10.21.4.15 stdair::CabinCode\_T AIRINV::DCPEventStruct::\_cabinCode**

Cabin code.

Definition at line 122 of file DCPEventStruct.hpp.

Referenced by describe().

**10.21.4.16 stdair::CityCode\_T AIRINV::DCPEventStruct::\_pos**

Point-of-sale.

Definition at line 125 of file DCPEventStruct.hpp.



Referenced by describe().

#### 10.21.4.17 stdair::ChannelLabel\_T AIRINV::DCPEventStruct::\_channel

Channel distribution.

Definition at line 128 of file DCPEventStruct.hpp.

Referenced by describe().

#### 10.21.4.18 stdair::DayDuration\_T AIRINV::DCPEventStruct::\_advancePurchase

Number of days that the ticket is sold before the flightDate.

Definition at line 131 of file DCPEventStruct.hpp.

Referenced by describe().

#### 10.21.4.19 stdair::SaturdayStay\_T AIRINV::DCPEventStruct::\_saturdayStay

Boolean saying whether a saturday is considered during the stay .

Definition at line 134 of file DCPEventStruct.hpp.

Referenced by describe().

#### 10.21.4.20 stdair::ChangeFees\_T AIRINV::DCPEventStruct::\_changeFees

Boolean saying whether the change fees option is requested or not.

Definition at line 137 of file DCPEventStruct.hpp.

Referenced by describe().

#### 10.21.4.21 stdair::NonRefundable\_T AIRINV::DCPEventStruct::\_nonRefundable

Boolean saying whether the refundable option is requested or not.

Definition at line 140 of file DCPEventStruct.hpp.

Referenced by describe().

#### 10.21.4.22 stdair::DayDuration\_T AIRINV::DCPEventStruct::\_minimumStay

Number of days that the customer spent into the destination city.

Definition at line 143 of file DCPEventStruct.hpp.

Referenced by describe().

#### 10.21.4.23 stdair::PriceValue\_T AIRINV::DCPEventStruct::\_DCP

Price.

Definition at line 146 of file DCPEventStruct.hpp.

Referenced by describe().

**10.21.4.24** `stdair::AirlineCode_T` [AIRINV::DCPEventStruct::\\_airlineCode](#)

Airline code

Definition at line 149 of file DCPEventStruct.hpp.

**10.21.4.25** `stdair::ClassCode_T` [AIRINV::DCPEventStruct::\\_classCode](#)

Code

Definition at line 152 of file DCPEventStruct.hpp.

**10.21.4.26** `stdair::AirlineCodeList_T` [AIRINV::DCPEventStruct::\\_airlineCodeList](#)

Airline Code List

Definition at line 155 of file DCPEventStruct.hpp.

Referenced by `beginAirline()`, `describe()`, `getAirlineListSize()`, `getCurrentAirlineCode()`, `getFirstAirlineCode()`, and `hasNotReachedEndAirline()`.

**10.21.4.27** `stdair::ClassList_StringList_T` [AIRINV::DCPEventStruct::\\_classCodeList](#)

Class Code List

Definition at line 161 of file DCPEventStruct.hpp.

Referenced by `beginClassCode()`, `describe()`, `getClassCodeListSize()`, `getCurrentClassCode()`, `getFirstClassCode()`, `hasNotReachedEndClassCode()`, `iterateAirline()`, and `iterateClassCode()`.

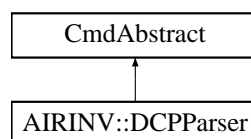
The documentation for this struct was generated from the following files:

- [airinv/bom/DCPEventStruct.hpp](#)
- [airinv/bom/DCPEventStruct.cpp](#)

**10.22 AIRINV::DCPParser Class Reference**

```
#include <airinv/command/vault/DCPParser.hpp>
```

Inheritance diagram for AIRINV::DCPParser::

**Static Public Member Functions**

- static void [DCPRuleGeneration](#) (const `stdair::Filename_T` &, `stdair::BomRoot` &)

**10.22.1 Detailed Description**

Class wrapping the parser entry point.

Definition at line 19 of file DCPParser.hpp.

### 10.22.2 Member Function Documentation

#### 10.22.2.1 void AIRINV::DCPParser::DCPRuleGeneration (const stdair::Filename\_T &, stdair::BomRoot &) [static]

Parses the CSV file describing the DCPs for the simulator, and generates the event structures accordingly.

##### Parameters:

- const* stdair::Filename\_T& The file-name of the CSV-formatted DCP input file.
- stdair::BomRoot&* Root of the BOM tree.

Definition at line 16 of file DCPParser.cpp.

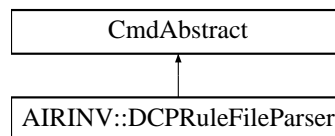
The documentation for this class was generated from the following files:

- [airinv/command/vault/DCPParser.hpp](#)
- [airinv/command/vault/DCPParser.cpp](#)

## 10.23 AIRINV::DCPRuleFileParser Class Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPRuleFileParser::



### Public Member Functions

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

#### 10.23.1 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 337 of file DCPParserHelper.hpp.

#### 10.23.2 Constructor & Destructor Documentation

##### 10.23.2.1 AIRINV::DCPRuleFileParser::DCPRuleFileParser (stdair::BomRoot & ioBomRoot, const stdair::Filename\_T & iFilename)

Constructor.

Definition at line 572 of file DCPParserHelper.cpp.

### 10.23.3 Member Function Documentation

#### 10.23.3.1 bool AIRINV::DCPRuleFileParser::generateDCPRules ()

Parse the input file and generate the Inventories.

Definition at line 593 of file DCPParserHelper.cpp.

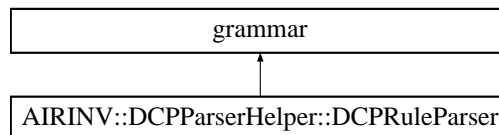
The documentation for this class was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.24 AIRINV::DCPParserHelper::DCPRuleParser Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::DCPRuleParser::



### Public Member Functions

- [DCPRuleParser](#) (stdair::BomRoot &, DCPRuleStruct &)

### Public Attributes

- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [start](#)
- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [comments](#)
- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [DCP\\_rule](#)
- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [DCP\\_rule\\_end](#)
- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [DCP\\_key](#)
- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [DCP\\_id](#)
- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [origin](#)
- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [destination](#)
- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [dateRangeStart](#)
- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [dateRangeEnd](#)
- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [date](#)
- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [timeRangeStart](#)
- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [timeRangeEnd](#)
- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [time](#)
- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [position](#)
- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [cabinCode](#)
- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [channel](#)
- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [advancePurchase](#)
- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [saturdayStay](#)
- boost::spirit::qi::rule< [stdair::iterator\\_t](#), boost::spirit::ascii::space\_type > [changeFees](#)

- `boost::spirit::qi::rule< stdair::iterator\_t, boost::spirit::ascii::space_type > nonRefundable`
- `boost::spirit::qi::rule< stdair::iterator\_t, boost::spirit::ascii::space_type > minimumStay`
- `boost::spirit::qi::rule< stdair::iterator\_t, boost::spirit::ascii::space_type > DCP`
- `boost::spirit::qi::rule< stdair::iterator\_t, boost::spirit::ascii::space_type > segment`
- `boost::spirit::qi::rule< stdair::iterator\_t, boost::spirit::ascii::space_type > list\_class`
- `stdair::BomRoot & \_bomRoot`
- `DCPRuleStruct & \_DCPRule`

### 10.24.1 Detailed Description

Grammar for the DCP-Rule parser.

Definition at line 304 of file `DCPParserHelper.hpp`.

### 10.24.2 Constructor & Destructor Documentation

#### 10.24.2.1 AIRINV::DCPParserHelper::DCPRuleParser::DCPRuleParser (stdair::BomRoot &, DCPRuleStruct &)

Definition at line 453 of file `DCPParserHelper.cpp`.

References `_bomRoot`, `_DCPRule`, `advancePurchase`, `cabinCode`, `changeFees`, `channel`, `comments`, `date`, `dateRangeEnd`, `dateRangeStart`, `AIRINV::DCPParserHelper::day_p`, `DCP`, `DCP_id`, `DCP_key`, `DCP_rule`, `DCP_rule_end`, `destination`, `AIRINV::DCPParserHelper::hour_p`, `list_class`, `minimumStay`, `AIRINV::DCPParserHelper::minute_p`, `AIRINV::DCPParserHelper::month_p`, `nonRefundable`, `origin`, `position`, `saturdayStay`, `AIRINV::DCPParserHelper::second_p`, `segment`, `start`, `time`, `timeRangeEnd`, `timeRangeStart`, `AIRINV::DCPParserHelper::uint1_4_p`, and `AIRINV::DCPParserHelper::year_p`.

### 10.24.3 Member Data Documentation

#### 10.24.3.1 boost::spirit::qi::rule<stdair::iterator\_t, AIRINV::DCPParserHelper::DCPRuleParser::start, boost::spirit::ascii::space\_type>

Definition at line 313 of file `DCPParserHelper.hpp`.

Referenced by `DCPRuleParser()`.

#### 10.24.3.2 boost::spirit::qi::rule<stdair::iterator\_t, AIRINV::DCPParserHelper::DCPRuleParser::comments, boost::spirit::ascii::space\_type>

Definition at line 313 of file `DCPParserHelper.hpp`.

Referenced by `DCPRuleParser()`.

#### 10.24.3.3 boost::spirit::qi::rule<stdair::iterator\_t, AIRINV::DCPParserHelper::DCPRuleParser::DCP\_rule, boost::spirit::ascii::space\_type>

Definition at line 313 of file `DCPParserHelper.hpp`.

Referenced by `DCPRuleParser()`.

#### 10.24.3.4 boost::spirit::qi::rule<stdair::iterator\_t, AIRINV::DCPParserHelper::DCPRuleParser::DCP\_rule\_end, boost::spirit::ascii::space\_type>

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.5** `boost::spirit::qi::rule<stdair::iterator_t,  
AIRINV::DCPParserHelper::DCPRuleParser::DCP_key`

`boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.6** `boost::spirit::qi::rule<stdair::iterator_t,  
AIRINV::DCPParserHelper::DCPRuleParser::DCP_id`

`boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.7** `boost::spirit::qi::rule<stdair::iterator_t,  
AIRINV::DCPParserHelper::DCPRuleParser::origin`

`boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.8** `boost::spirit::qi::rule<stdair::iterator_t,  
AIRINV::DCPParserHelper::DCPRuleParser::destination`

`boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.9** `boost::spirit::qi::rule<stdair::iterator_t,  
AIRINV::DCPParserHelper::DCPRuleParser::dateRangeStart`

`boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.10** `boost::spirit::qi::rule<stdair::iterator_t,  
AIRINV::DCPParserHelper::DCPRuleParser::dateRangeEnd`

`boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.11** `boost::spirit::qi::rule<stdair::iterator_t,  
AIRINV::DCPParserHelper::DCPRuleParser::date`

`boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.12** `boost::spirit::qi::rule<stdair::iterator_t,  
AIRINV::DCPParserHelper::DCPRuleParser::timeRangeStart`

`boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.13** `boost::spirit::qi::rule<stdair::iterator_t, AIRINV::DCPParserHelper::DCPRuleParser::timeRangeEnd, boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.14** `boost::spirit::qi::rule<stdair::iterator_t, AIRINV::DCPParserHelper::DCPRuleParser::time, boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.15** `boost::spirit::qi::rule<stdair::iterator_t, AIRINV::DCPParserHelper::DCPRuleParser::position, boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.16** `boost::spirit::qi::rule<stdair::iterator_t, AIRINV::DCPParserHelper::DCPRuleParser::cabinCode, boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.17** `boost::spirit::qi::rule<stdair::iterator_t, AIRINV::DCPParserHelper::DCPRuleParser::channel, boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.18** `boost::spirit::qi::rule<stdair::iterator_t, AIRINV::DCPParserHelper::DCPRuleParser::advancePurchase, boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.19** `boost::spirit::qi::rule<stdair::iterator_t, AIRINV::DCPParserHelper::DCPRuleParser::saturdayStay, boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.20** `boost::spirit::qi::rule<stdair::iterator_t, AIRINV::DCPParserHelper::DCPRuleParser::changeFees, boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.21** `boost::spirit::qi::rule<stdair::iterator_t, AIRINV::DCPParserHelper::DCPRuleParser::nonRefundable, boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.22** `boost::spirit::qi::rule<stdair::iterator_t, AIRINV::DCPParserHelper::DCPRuleParser::minimumStay, boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.23** `boost::spirit::qi::rule<stdair::iterator_t, AIRINV::DCPParserHelper::DCPRuleParser::DCP, boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.24** `boost::spirit::qi::rule<stdair::iterator_t, AIRINV::DCPParserHelper::DCPRuleParser::segment, boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.25** `boost::spirit::qi::rule<stdair::iterator_t, AIRINV::DCPParserHelper::DCPRuleParser::list_class, boost::spirit::ascii::space_type>`

Definition at line 313 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.26** `stdair::BomRoot& AIRINV::DCPParserHelper::DCPRuleParser::_bomRoot`

Definition at line 320 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

**10.24.3.27** `DCPRuleStruct& AIRINV::DCPParserHelper::DCPRuleParser::_DCPRule`

Definition at line 321 of file DCPParserHelper.hpp.

Referenced by DCPRuleParser().

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

- `airinv/command/vault/DCPParserHelper.hpp`
- `airinv/command/vault/DCPParserHelper.cpp`

## 10.25 AIRINV::DefaultMap Struct Reference

```
#include <airinv/basic/BasConst_Curves.hpp>
```



### Static Public Member Functions

- static [FRAT5Curve\\_T createPickupFRAT5Curve \(\)](#)

#### 10.25.1 Detailed Description

Default PoS probability mass.

Definition at line 16 of file BasConst\_Curves.hpp.

#### 10.25.2 Member Function Documentation

##### 10.25.2.1 [FRAT5Curve\\_T](#) AIRINV::DefaultMap::createPickupFRAT5Curve () [static]

Definition at line 16 of file BasConst.cpp.

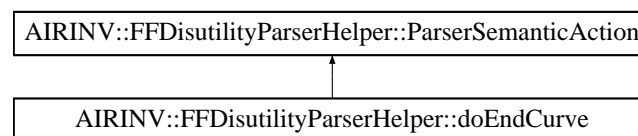
The documentation for this struct was generated from the following files:

- [airinv/basic/BasConst\\_Curves.hpp](#)
- [airinv/basic/BasConst.cpp](#)

## 10.26 AIRINV::FFDisutilityParserHelper::doEndCurve Struct Reference

```
#include <airinv/command/FFDisutilityParserHelper.hpp>
```

Inheritance diagram for AIRINV::FFDisutilityParserHelper::doEndCurve::



### Public Member Functions

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

### Public Attributes

- stdair::BomRoot & [\\_bomRoot](#)
- [FFDisutilityStruct](#) & [\\_ffDisutility](#)

#### 10.26.1 Detailed Description

Mark the end of the FFDisutility curve parsing.

Definition at line 61 of file FFDisutilityParserHelper.hpp.

## 10.26.2 Constructor & Destructor Documentation

### 10.26.2.1 AIRINV::FFDisutilityParserHelper::doEndCurve::doEndCurve (stdair::BomRoot &, [FFDisutilityStruct](#) &)

Actor Constructor.

Definition at line 79 of file [FFDisutilityParserHelper.cpp](#).

## 10.26.3 Member Function Documentation

### 10.26.3.1 void AIRINV::FFDisutilityParserHelper::doEndCurve::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 87 of file [FFDisutilityParserHelper.cpp](#).

References [\\_bomRoot](#), [AIRINV::FFDisutilityStruct::\\_curve](#), [AIRINV::FFDisutilityParserHelper::ParserSemanticAction::\\_ffDisutility](#), [AIRINV::FFDisutilityStruct::\\_key](#), and [AIRINV::FFDisutilityStruct::describe\(\)](#).

## 10.26.4 Member Data Documentation

### 10.26.4.1 stdair::BomRoot& [AIRINV::FFDisutilityParserHelper::doEndCurve::\\_bomRoot](#)

Actor Specific Context.

Definition at line 67 of file [FFDisutilityParserHelper.hpp](#).

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

### 10.26.4.2 [FFDisutilityStruct&](#) [AIRINV::FFDisutilityParserHelper::ParserSemanticAction::\\_ffDisutility](#) [inherited]

Actor Context.

Definition at line 33 of file [FFDisutilityParserHelper.hpp](#).

Referenced by [operator\(\)](#), [AIRINV::FFDisutilityParserHelper::storeFFDisutilityValue::operator\(\)](#), [AIRINV::FFDisutilityParserHelper::storeDTD::operator\(\)](#), and [AIRINV::FFDisutilityParserHelper::storeCurveKey::operator\(\)](#).

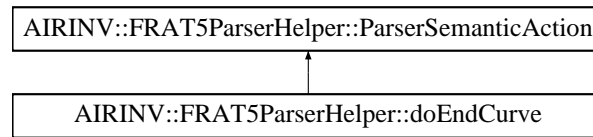
The documentation for this struct was generated from the following files:

- [airinv/command/FFDisutilityParserHelper.hpp](#)
- [airinv/command/FFDisutilityParserHelper.cpp](#)

## 10.27 AIRINV::FRAT5ParserHelper::doEndCurve Struct Reference

```
#include <airinv/command/FRAT5ParserHelper.hpp>
```

Inheritance diagram for [AIRINV::FRAT5ParserHelper::doEndCurve](#):



## Public Member Functions

- [doEndCurve](#) (stdair::BomRoot &, [FRAT5Struct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

## Public Attributes

- stdair::BomRoot & [\\_bomRoot](#)
- [FRAT5Struct](#) & [\\_frat5](#)

### 10.27.1 Detailed Description

Mark the end of the FRAT5 curve parsing.

Definition at line 61 of file FRAT5ParserHelper.hpp.

### 10.27.2 Constructor & Destructor Documentation

#### 10.27.2.1 AIRINV::FRAT5ParserHelper::doEndCurve::doEndCurve (stdair::BomRoot &, [FRAT5Struct](#) &)

Actor Constructor.

Definition at line 80 of file FRAT5ParserHelper.cpp.

### 10.27.3 Member Function Documentation

#### 10.27.3.1 void AIRINV::FRAT5ParserHelper::doEndCurve::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 88 of file FRAT5ParserHelper.cpp.

References [\\_bomRoot](#), [AIRINV::FRAT5Struct::\\_curve](#), [AIRINV::FRAT5ParserHelper::ParserSemanticAction::\\_frat5](#), [AIRINV::FRAT5Struct::\\_key](#), and [AIRINV::FRAT5Struct::describe\(\)](#).

### 10.27.4 Member Data Documentation

#### 10.27.4.1 stdair::BomRoot& [AIRINV::FRAT5ParserHelper::doEndCurve::\\_bomRoot](#)

Actor Specific Context.

Definition at line 67 of file FRAT5ParserHelper.hpp.

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

#### 10.27.4.2 FRAT5Struct& AIRINV::FRAT5ParserHelper::ParserSemanticAction::\_frat5 [inherited]

Actor Context.

Definition at line 33 of file FRAT5ParserHelper.hpp.

Referenced by operator(), AIRINV::FRAT5ParserHelper::storeFRAT5Value::operator(), AIRINV::FRAT5ParserHelper::storeDTD::operator(), and AIRINV::FRAT5ParserHelper::storeCurveKey::operator().

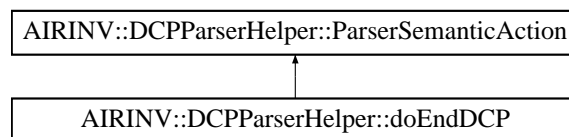
The documentation for this struct was generated from the following files:

- [airinv/command/FRAT5ParserHelper.hpp](#)
- [airinv/command/FRAT5ParserHelper.cpp](#)

## 10.28 AIRINV::DCPParserHelper::doEndDCP Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::doEndDCP::



### Public Member Functions

- [doEndDCP](#) (stdair::BomRoot &, DCPRuleStruct &)
- void [operator\(\)](#) (boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

### Public Attributes

- stdair::BomRoot & [\\_bomRoot](#)
- DCPRuleStruct & [\\_DCPRule](#)

#### 10.28.1 Detailed Description

Mark the end of the DCP-rule parsing.

Definition at line 218 of file DCPParserHelper.hpp.

#### 10.28.2 Constructor & Destructor Documentation

##### 10.28.2.1 AIRINV::DCPParserHelper::doEndDCP::doEndDCP (stdair::BomRoot &, DCPRuleStruct &)

Actor Constructor.

Definition at line 399 of file DCPParserHelper.cpp.

### 10.28.3 Member Function Documentation

#### 10.28.3.1 void AIRINV::DCPParserHelper::doEndDCP::operator() (boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

Actor Function (functor).

Definition at line 406 of file DCPParserHelper.cpp.

References `_bomRoot`, and `AIRINV::DCPParserHelper::ParserSemanticAction::_DCPRule`.

### 10.28.4 Member Data Documentation

#### 10.28.4.1 stdair::BomRoot& AIRINV::DCPParserHelper::doEndDCP::\_bomRoot

Actor Specific Context.

Definition at line 226 of file DCPParserHelper.hpp.

Referenced by `operator()`.

#### 10.28.4.2 DCPRuleStruct& AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule [inherited]

Actor Context.

Definition at line 34 of file DCPParserHelper.hpp.

Referenced by `operator()`, `AIRINV::DCPParserHelper::storeClass::operator()`, `AIRINV::DCPParserHelper::storeAirlineCode::operator()`, `AIRINV::DCPParserHelper::storeDCP::operator()`, `AIRINV::DCPParserHelper::storeMinimumStay::operator()`, `AIRINV::DCPParserHelper::storeNonRefundable::operator()`, `AIRINV::DCPParserHelper::storeChangeFees::operator()`, `AIRINV::DCPParserHelper::storeSaturdayStay::operator()`, `AIRINV::DCPParserHelper::storeAdvancePurchase::operator()`, `AIRINV::DCPParserHelper::storeChannel::operator()`, `AIRINV::DCPParserHelper::storeCabinCode::operator()`, `AIRINV::DCPParserHelper::storePOS::operator()`, `AIRINV::DCPParserHelper::storeEndRangeTime::operator()`, `AIRINV::DCPParserHelper::storeStartRangeTime::operator()`, `AIRINV::DCPParserHelper::storeDateRangeEnd::operator()`, `AIRINV::DCPParserHelper::storeDateRangeStart::operator()`, `AIRINV::DCPParserHelper::storeDestination::operator()`, `AIRINV::DCPParserHelper::storeOrigin::operator()`, and `AIRINV::DCPParserHelper::storeDCPID::operator()`.

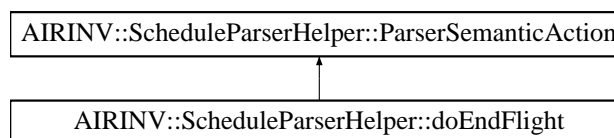
The documentation for this struct was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.29 AIRINV::ScheduleParserHelper::doEndFlight Struct Reference

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

Inheritance diagram for `AIRINV::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](#)

#### 10.29.1 Detailed Description

Mark the end of the flight-period parsing.

Definition at line 224 of file ScheduleParserHelper.hpp.

#### 10.29.2 Constructor & Destructor Documentation

##### 10.29.2.1 AIRINV::ScheduleParserHelper::doEndFlight::doEndFlight (stdair::BomRoot &, [FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 436 of file ScheduleParserHelper.cpp.

#### 10.29.3 Member Function Documentation

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

Actor Function (functor).

Definition at line 444 of file ScheduleParserHelper.cpp.

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

#### 10.29.4 Member Data Documentation

##### 10.29.4.1 stdair::BomRoot& AIRINV::ScheduleParserHelper::doEndFlight::\_bomRoot

Actor Specific Context.

Definition at line 230 of file ScheduleParserHelper.hpp.

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

##### 10.29.4.2 [FlightPeriodStruct](#)& AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flight-Period [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

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

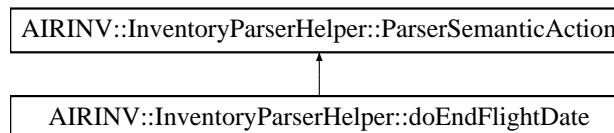
The documentation for this struct was generated from the following files:

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

## 10.30 AIRINV::InventoryParserHelper::doEndFlightDate Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::doEndFlightDate::



### Public Member Functions

- [doEndFlightDate](#) (stdair::BomRoot & [FlightDateStruct](#) &, unsigned int &)
- void [operator\(\)](#) (iterator\_t iStr, iterator\_t iStrEnd) const

### Public Attributes

- stdair::BomRoot & [\\_bomRoot](#)
- unsigned int & [\\_nbOfFlights](#)
- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.30.1 Detailed Description

Mark the end of the inventory parsing.

Definition at line 441 of file InventoryParserHelper.hpp.

### 10.30.2 Constructor & Destructor Documentation

#### 10.30.2.1 AIRINV::InventoryParserHelper::doEndFlightDate::doEndFlightDate (stdair::BomRoot &, FlightDateStruct &, unsigned int &)

Actor Constructor.

Definition at line 778 of file InventoryParserHelper.cpp.

### 10.30.3 Member Function Documentation

#### 10.30.3.1 void AIRINV::InventoryParserHelper::doEndFlightDate::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 787 of file InventoryParserHelper.cpp.

References `_bomRoot`, `AIRINV::SegmentStruct::_cabinList`, `AIRINV::InventoryParserHelper::ParserSemanticAction::_flightDate`, `AIRINV::FlightDateStruct::_itSegment`, `_nbOfFlights`, and `AIRINV::FlightDateStruct::_segmentList`.

### 10.30.4 Member Data Documentation

#### 10.30.4.1 stdair::BomRoot& AIRINV::InventoryParserHelper::doEndFlightDate::\_bomRoot

Actor Specific Context.

Definition at line 448 of file InventoryParserHelper.hpp.

Referenced by `operator()`.

#### 10.30.4.2 unsigned int& AIRINV::InventoryParserHelper::doEndFlightDate::\_nbOfFlights

Definition at line 449 of file InventoryParserHelper.hpp.

Referenced by `operator()`.

#### 10.30.4.3 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by `operator()`, `AIRINV::InventoryParserHelper::storeFCClasses::operator()`, `AIRINV::InventoryParserHelper::storeFamilyCode::operator()`, `AIRINV::InventoryParserHelper::storeRevenueAvailability::operator()`, `AIRINV::InventoryParserHelper::storeSegmentAvailability::operator()`, `AIRINV::InventoryParserHelper::storeClassAvailability::operator()`, `AIRINV::InventoryParserHelper::storeClassETB::operator()`, `AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator()`, `AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator()`, `AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator()`, `AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator()`, `AIRINV::InventoryParserHelper::storeOverbooking::operator()`, `AIRINV::InventoryParserHelper::storeNoShow::operator()`, `AIRINV::InventoryParserHelper::storeNego::operator()`, `AIRINV::InventoryParserHelper::storeProtection::operator()`, `AIRINV::InventoryParserHelper::storeCumulatedProtection::operator()`, and `AIRINV::InventoryParserHelper::storeParentSubclassCode::operator()`, `AIRINV::InventoryParserHelper::storeParentClassCode::operator()`.



AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

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

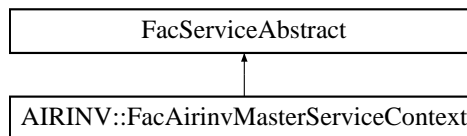
- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.31 AIRINV::FacAirinvMasterServiceContext Class Reference

Factory for Bucket.

```
#include <airinv/factory/FacAirinvMasterServiceContext.hpp>
```

Inheritance diagram for AIRINV::FacAirinvMasterServiceContext::



### Public Member Functions

- [~FacAirinvMasterServiceContext \(\)](#)
- [AIRINV\\_Master\\_ServiceContext & create \(\)](#)

### Static Public Member Functions

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

## Protected Member Functions

- [FacAirinvMasterServiceContext](#) ()

### 10.31.1 Detailed Description

Factory for Bucket.

Definition at line 20 of file FacAirinvMasterServiceContext.hpp.

### 10.31.2 Constructor & Destructor Documentation

#### 10.31.2.1 AIRINV::FacAirinvMasterServiceContext::~~FacAirinvMasterServiceContext ()

Destructor.

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

Definition at line 17 of file FacAirinvMasterServiceContext.cpp.

#### 10.31.2.2 AIRINV::FacAirinvMasterServiceContext::FacAirinvMasterServiceContext () [inline, protected]

Default Constructor.

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

Definition at line 44 of file FacAirinvMasterServiceContext.hpp.

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

### 10.31.3 Member Function Documentation

#### 10.31.3.1 [FacAirinvMasterServiceContext](#) & AIRINV::FacAirinvMasterServiceContext::instance () [static]

Provide the unique instance.

The singleton is instantiated when first used

#### Returns:

[FacAirinvMasterServiceContext](#)&

Definition at line 22 of file FacAirinvMasterServiceContext.cpp.

References [FacAirinvMasterServiceContext\(\)](#).

#### 10.31.3.2 [AIRINV\\_Master\\_ServiceContext](#) & AIRINV::FacAirinvMasterServiceContext::create ()

Create a new [AIRINV\\_Master\\_ServiceContext](#) object.

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

#### Returns:

[AIRINV\\_Master\\_ServiceContext](#)& The newly created object.

Definition at line 34 of file FacAirinvMasterServiceContext.cpp.

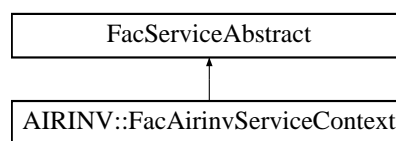
The documentation for this class was generated from the following files:

- [airinv/factory/FacAirinvMasterServiceContext.hpp](#)
- [airinv/factory/FacAirinvMasterServiceContext.cpp](#)

## 10.32 AIRINV::FacAirinvServiceContext Class Reference

```
#include <airinv/factory/FacAirinvServiceContext.hpp>
```

Inheritance diagram for AIRINV::FacAirinvServiceContext::



### Public Member Functions

- [~FacAirinvServiceContext \(\)](#)
- [AIRINV\\_ServiceContext & create \(\)](#)

### Static Public Member Functions

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

### Protected Member Functions

- [FacAirinvServiceContext \(\)](#)

#### 10.32.1 Detailed Description

Factory for Bucket.

Definition at line 18 of file FacAirinvServiceContext.hpp.

#### 10.32.2 Constructor & Destructor Documentation

##### 10.32.2.1 AIRINV::FacAirinvServiceContext::~~FacAirinvServiceContext ()

Destructor.

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

Definition at line 17 of file FacAirinvServiceContext.cpp.

**10.32.2.2 AIRINV::FacAirinvServiceContext::FacAirinvServiceContext** () [inline, protected]

Default Constructor.

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

Definition at line 42 of file FacAirinvServiceContext.hpp.

Referenced by instance().

### 10.32.3 Member Function Documentation

**10.32.3.1 FacAirinvServiceContext & AIRINV::FacAirinvServiceContext::instance ()** [static]

Provide the unique instance.

The singleton is instantiated when first used

#### Returns:

[FacAirinvServiceContext&](#)

Definition at line 22 of file FacAirinvServiceContext.cpp.

References [FacAirinvServiceContext\(\)](#).

**10.32.3.2 AIRINV\_ServiceContext & AIRINV::FacAirinvServiceContext::create ()**

Create a new [AIRINV\\_ServiceContext](#) object.

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

#### Returns:

[AIRINV\\_ServiceContext&](#) The newly created object.

Definition at line 34 of file FacAirinvServiceContext.cpp.

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

- [airinv/factory/FacAirinvServiceContext.hpp](#)
- [airinv/factory/FacAirinvServiceContext.cpp](#)

## 10.33 AIRINV::FacBomAbstract Class Reference

```
#include <airinv/factory/FacBomAbstract.hpp>
```

### Public Types

- typedef std::vector< [BomAbstract](#) \* > [BomPool\\_T](#)

### Static Public Member Functions

- static std::size\_t [getID](#) (const [BomAbstract](#) \*)

- static std::size\_t [getID](#) (const [BomAbstract](#) &)
- static std::string [getIDString](#) (const [BomAbstract](#) \*)
- static std::string [getIDString](#) (const [BomAbstract](#) &)

### Protected Member Functions

- [FacBomAbstract](#) ()
- [FacBomAbstract](#) (const [FacBomAbstract](#) &)
- virtual [~FacBomAbstract](#) ()

### Protected Attributes

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

### Friends

- class [FacSupervisor](#)

#### 10.33.1 Detailed Description

Base class for Factory layer.

Definition at line 17 of file [FacBomAbstract.hpp](#).

#### 10.33.2 Member Typedef Documentation

##### 10.33.2.1 `typedef std::vector<BomAbstract*> AIRINV::FacBomAbstract::BomPool_T`

Define the list (pool) of Bom objects.

Definition at line 22 of file [FacBomAbstract.hpp](#).

#### 10.33.3 Constructor & Destructor Documentation

##### 10.33.3.1 `AIRINV::FacBomAbstract::FacBomAbstract () [inline, protected]`

Default Constructor.

This constructor is protected to ensure the class is abstract.

Definition at line 41 of file [FacBomAbstract.hpp](#).

##### 10.33.3.2 `AIRINV::FacBomAbstract::FacBomAbstract (const FacBomAbstract &) [inline, protected]`

Definition at line 42 of file [FacBomAbstract.hpp](#).

##### 10.33.3.3 `AIRINV::FacBomAbstract::~~FacBomAbstract () [protected, virtual]`

Destructor.

Definition at line 16 of file [FacBomAbstract.cpp](#).

### 10.33.4 Member Function Documentation

#### 10.33.4.1 `std::size_t AIRINV::FacBomAbstract::getID (const BomAbstract *)` [static]

Return the ID corresponding to the given object pointer.

Definition at line 35 of file FacBomAbstract.cpp.

Referenced by `getID()`, and `getIDString()`.

#### 10.33.4.2 `std::size_t AIRINV::FacBomAbstract::getID (const BomAbstract &)` [static]

Return the ID corresponding to the given object reference.

Definition at line 43 of file FacBomAbstract.cpp.

References `getID()`.

#### 10.33.4.3 `std::string AIRINV::FacBomAbstract::getIDString (const BomAbstract *)` [static]

Return the ID, as a string, corresponding to the given object pointer.

Definition at line 48 of file FacBomAbstract.cpp.

References `getID()`.

Referenced by `getIDString()`.

#### 10.33.4.4 `std::string AIRINV::FacBomAbstract::getIDString (const BomAbstract &)` [static]

Return the ID, as a string, corresponding to the given object reference.

Definition at line 56 of file FacBomAbstract.cpp.

References `getIDString()`.

### 10.33.5 Friends And Related Function Documentation

#### 10.33.5.1 `friend class FacSupervisor` [friend]

Definition at line 18 of file FacBomAbstract.hpp.

### 10.33.6 Member Data Documentation

#### 10.33.6.1 `BomPool_T AIRINV::FacBomAbstract::_pool` [protected]

List of instantiated Business Objects

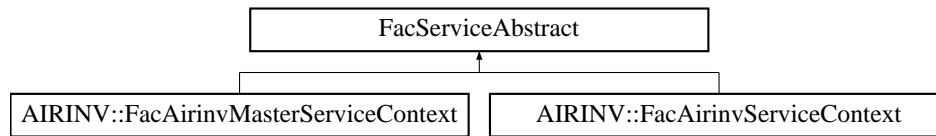
Definition at line 53 of file FacBomAbstract.hpp.

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

- [airinv/factory/FacBomAbstract.hpp](#)
- [airinv/factory/FacBomAbstract.cpp](#)

## 10.34 FacServiceAbstract Class Reference

Inheritance diagram for FacServiceAbstract::



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

- [airinv/factory/FacAirinvMasterServiceContext.hpp](#)
- [airinv/factory/FacAirinvServiceContext.hpp](#)

## 10.35 AIRINV::FacServiceAbstract Class Reference

```
#include <airinv/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`

### 10.35.1 Detailed Description

Base class for the (Service) Factory layer.

Definition at line 16 of file `FacServiceAbstract.hpp`.

### 10.35.2 Member Typedef Documentation

#### 10.35.2.1 `typedef std::vector<ServiceAbstract*> AIRINV::FacServiceAbstract::ServicePool\_T`

Define the list (pool) of Service objects.

Definition at line 20 of file `FacServiceAbstract.hpp`.

### 10.35.3 Constructor & Destructor Documentation

#### 10.35.3.1 AIRINV::FacServiceAbstract::~~FacServiceAbstract () [virtual]

Destructor.

Definition at line 13 of file FacServiceAbstract.cpp.

#### 10.35.3.2 AIRINV::FacServiceAbstract::FacServiceAbstract () [inline, protected]

Default Constructor.

This constructor is protected to ensure the class is abstract.

Definition at line 31 of file FacServiceAbstract.hpp.

### 10.35.4 Member Function Documentation

#### 10.35.4.1 void AIRINV::FacServiceAbstract::clean ()

Destroyed all the object instantiated by this factory.

Definition at line 18 of file FacServiceAbstract.cpp.

### 10.35.5 Member Data Documentation

#### 10.35.5.1 [ServicePool\\_T](#) AIRINV::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:

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

## 10.36 AIRINV::FacSupervisor Class Reference

```
#include <airinv/factory/FacSupervisor.hpp>
```

### Public Types

- typedef std::vector< [FacBomAbstract \\*](#) > [BomFactoryPool\\_T](#)
- typedef std::vector< [FacServiceAbstract \\*](#) > [ServiceFactoryPool\\_T](#)

### Public Member Functions

- void [registerBomFactory](#) ([FacBomAbstract \\*](#))
- void [registerServiceFactory](#) ([FacServiceAbstract \\*](#))
- void [cleanBomLayer](#) ()
- void [cleanServiceLayer](#) ()
- [~FacSupervisor](#) ()



### Static Public Member Functions

- static [FacSupervisor](#) & [instance](#) ()
- static void [cleanFactory](#) ()

### Protected Member Functions

- [FacSupervisor](#) ()
- [FacSupervisor](#) (const [FacSupervisor](#) &)

#### 10.36.1 Detailed Description

Singleton class to register and clean all Factories.

Definition at line 17 of file FacSupervisor.hpp.

#### 10.36.2 Member Typedef Documentation

##### 10.36.2.1 typedef std::vector<[FacBomAbstract\\*](#)> [AIRINV::FacSupervisor::BomFactoryPool\\_T](#)

Define the pool (list) of factories.

Definition at line 21 of file FacSupervisor.hpp.

##### 10.36.2.2 typedef std::vector<[FacServiceAbstract\\*](#)> [AIRINV::FacSupervisor::ServiceFactoryPool\\_T](#)

Definition at line 22 of file FacSupervisor.hpp.

#### 10.36.3 Constructor & Destructor Documentation

##### 10.36.3.1 AIRINV::FacSupervisor::~~FacSupervisor ()

Destructor

The static instance is deleted (and reset to NULL) by the static [cleanFactory\(\)](#) method.

Definition at line 41 of file FacSupervisor.cpp.

References [cleanBomLayer\(\)](#), and [cleanServiceLayer\(\)](#).

##### 10.36.3.2 AIRINV::FacSupervisor::FacSupervisor () [protected]

Default Constructor.

This constructor is protected to ensure the singleton pattern.

Definition at line 16 of file FacSupervisor.cpp.

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

##### 10.36.3.3 AIRINV::FacSupervisor::FacSupervisor (const [FacSupervisor](#) &) [inline, protected]

Definition at line 66 of file FacSupervisor.hpp.

### 10.36.4 Member Function Documentation

#### 10.36.4.1 [FacSupervisor](#) & AIRINV::FacSupervisor::instance () [static]

Provides the unique instance.

The singleton is instantiated when first used.

**Returns:**

[FacSupervisor](#)&

Definition at line 20 of file FacSupervisor.cpp.

References [FacSupervisor\(\)](#).

#### 10.36.4.2 void AIRINV::FacSupervisor::registerBomFactory ([FacBomAbstract](#) \*)

Register a newly instantiated concrete factory for the Bom layer.

When a concrete Factory is firstly instantiated this factory have to register itself to the [FacSupervisor](#)

**Parameters:**

*FacAbstract&* the concrete Factory to register.

Definition at line 30 of file FacSupervisor.cpp.

#### 10.36.4.3 void AIRINV::FacSupervisor::registerServiceFactory ([FacServiceAbstract](#) \*)

Register a newly instantiated concrete factory for the Service layer.

When a concrete Factory is firstly instantiated this factory have to register itself to the [FacSupervisor](#).

**Parameters:**

*FacServiceAbstract&* the concrete Factory to register.

Definition at line 36 of file FacSupervisor.cpp.

#### 10.36.4.4 void AIRINV::FacSupervisor::cleanBomLayer ()

Clean all created object.

Call the clean method of all the instantiated factories for the Bom layer.

Definition at line 47 of file FacSupervisor.cpp.

Referenced by [cleanFactory\(\)](#), and [~FacSupervisor\(\)](#).

#### 10.36.4.5 void AIRINV::FacSupervisor::cleanServiceLayer ()

Clean all Service created object.

Call the clean method of all the instantiated factories for the Service layer.

Definition at line 61 of file FacSupervisor.cpp.

Referenced by [cleanFactory\(\)](#), and [~FacSupervisor\(\)](#).

**10.36.4.6 void AIRINV::FacSupervisor::cleanFactory () [static]**

Clean the static instance.

The singleton is deleted.

Definition at line 75 of file FacSupervisor.cpp.

References `cleanBomLayer()`, and `cleanServiceLayer()`.

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

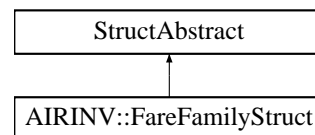
- [airinv/factory/FacSupervisor.hpp](#)
- [airinv/factory/FacSupervisor.cpp](#)

**10.37 AIRINV::FareFamilyStruct Struct Reference**

Utility Structure for the parsing of fare family details.

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

Inheritance diagram for AIRINV::FareFamilyStruct::

**Public Member Functions**

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

**Public Attributes**

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

**10.37.1 Detailed Description**

Utility Structure for the parsing of fare family details.

Definition at line 26 of file FareFamilyStruct.hpp.

## 10.37.2 Constructor & Destructor Documentation

### 10.37.2.1 AIRINV::FareFamilyStruct::FareFamilyStruct ()

Default constructor.

Definition at line 16 of file FareFamilyStruct.cpp.

### 10.37.2.2 AIRINV::FareFamilyStruct::FareFamilyStruct (const stdair::FamilyCode\_T &, const stdair::CurveKey\_T &, const stdair::CurveKey\_T &, const stdair::ClassList\_String\_T &)

Main constructor.

Definition at line 23 of file FareFamilyStruct.cpp.

## 10.37.3 Member Function Documentation

### 10.37.3.1 void AIRINV::FareFamilyStruct::fill (stdair::FareFamily &) const

Fill the FareFamily objects with the attributes of the [FareFamilyStruct](#).

Definition at line 52 of file FareFamilyStruct.cpp.

### 10.37.3.2 const std::string AIRINV::FareFamilyStruct::describe () const

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

Definition at line 32 of file FareFamilyStruct.cpp.

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

## 10.37.4 Member Data Documentation

### 10.37.4.1 stdair::FamilyCode\_T AIRINV::FareFamilyStruct::\_familyCode

Definition at line 28 of file FareFamilyStruct.hpp.

Referenced by [describe\(\)](#), [AIRINV::ScheduleParserHelper::storeFClasses::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeFamilyCode::operator\(\)](#), and [AIRINV::InventoryParserHelper::storeFamilyCode::operator\(\)](#).

### 10.37.4.2 stdair::CurveKey\_T AIRINV::FareFamilyStruct::\_frat5CurveKey

Definition at line 29 of file FareFamilyStruct.hpp.

Referenced by [describe\(\)](#), [AIRINV::ScheduleParserHelper::storeFClasses::operator\(\)](#), and [AIRINV::ScheduleParserHelper::storeFRAT5CurveKey::operator\(\)](#).

### 10.37.4.3 stdair::CurveKey\_T AIRINV::FareFamilyStruct::\_ffDisutilityCurveKey

Definition at line 30 of file FareFamilyStruct.hpp.

Referenced by [describe\(\)](#), [AIRINV::ScheduleParserHelper::storeFClasses::operator\(\)](#), and [AIRINV::ScheduleParserHelper::storeFFDisutilityCurveKey::operator\(\)](#).

### 10.37.4.4 stdair::ClassList\_String\_T AIRINV::FareFamilyStruct::\_classes

Definition at line 31 of file FareFamilyStruct.hpp.

Referenced by describe(), AIRINV::ScheduleParserHelper::storeClasses::operator>(), and AIRINV::InventoryParserHelper::storeFCClasses::operator>().

#### 10.37.4.5 BookingClassStructList\_T AIRINV::FareFamilyStruct::\_classList

Definition at line 32 of file FareFamilyStruct.hpp.

Referenced by describe(), AIRINV::InventoryParserHelper::storeFCClasses::operator>(), AIRINV::InventoryParserHelper::storeClassCode::operator>(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator>(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator>(), and AIRINV::InventoryParserHelper::storeAirlineCode::operator>().

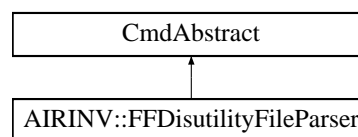
The documentation for this struct was generated from the following files:

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

## 10.38 AIRINV::FFDisutilityFileParser Class Reference

```
#include <airinv/command/FFDisutilityParserHelper.hpp>
```

Inheritance diagram for AIRINV::FFDisutilityFileParser::



### Public Member Functions

- [FFDisutilityFileParser](#) (stdair::BomRoot &ioBomRoot, const stdair::Filename\_T &iFilename)
- [generateFFDisutilityCurves](#) ()

#### 10.38.1 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 126 of file FFDisutilityParserHelper.hpp.

#### 10.38.2 Constructor & Destructor Documentation

##### 10.38.2.1 AIRINV::FFDisutilityFileParser::FFDisutilityFileParser (stdair::BomRoot & ioBomRoot, const stdair::Filename\_T & iFilename)

Constructor.

Definition at line 177 of file FFDisutilityParserHelper.cpp.

### 10.38.3 Member Function Documentation

#### 10.38.3.1 bool AIRINV::FFDisutilityFileParser::generateFFDisutilityCurves ()

Parse the input file and generate the FFDisutility curves.

Definition at line 201 of file FFDisutilityParserHelper.cpp.

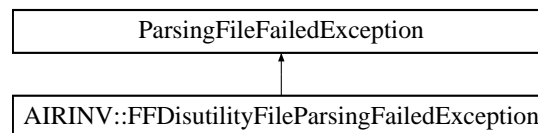
The documentation for this class was generated from the following files:

- [airinv/command/FFDisutilityParserHelper.hpp](#)
- [airinv/command/FFDisutilityParserHelper.cpp](#)

## 10.39 AIRINV::FFDisutilityFileParsingFailedException Class Reference

```
#include <airinv/AIRINV_Types.hpp>
```

Inheritance diagram for AIRINV::FFDisutilityFileParsingFailedException::



### Public Member Functions

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

#### 10.39.1 Detailed Description

The FFDisutility input file can not be parsed.

Definition at line 80 of file AIRINV\_Types.hpp.

### 10.39.2 Constructor & Destructor Documentation

#### 10.39.2.1 AIRINV::FFDisutilityFileParsingFailedException::FFDisutilityFileParsingFailed-Exception (const std::string &*iWhat*) [inline]

Constructor.

Definition at line 86 of file AIRINV\_Types.hpp.

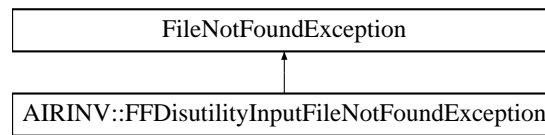
The documentation for this class was generated from the following file:

- [airinv/AIRINV\\_Types.hpp](#)

## 10.40 AIRINV::FFDisutilityInputFileNotFoundedException Class Reference

```
#include <airinv/AIRINV_Types.hpp>
```

Inheritance diagram for AIRINV::FFDisutilityInputFileNotFoundedException::



### Public Member Functions

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

#### 10.40.1 Detailed Description

The FF disutility input file can not be found or opened.

Definition at line 142 of file AIRINV\_Types.hpp.

#### 10.40.2 Constructor & Destructor Documentation

##### 10.40.2.1 AIRINV::FFDisutilityInputFileNotFoundException::FFDisutilityInputFileNotFoundException (const std::string & *iWhat*) [inline]

Constructor.

Definition at line 147 of file AIRINV\_Types.hpp.

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

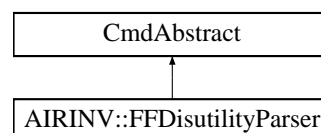
- [airinv/AIRINV\\_Types.hpp](#)

## 10.41 AIRINV::FFDisutilityParser Class Reference

Class wrapping the parser entry point.

```
#include <airinv/command/FFDisutilityParser.hpp>
```

Inheritance diagram for AIRINV::FFDisutilityParser::



### Static Public Member Functions

- static void [parse](#) (const stdair::FFDisutilityFilePath &iFFDisutilityInputFilename, stdair::BomRoot &)

#### 10.41.1 Detailed Description

Class wrapping the parser entry point.

Definition at line 22 of file FFDisutilityParser.hpp.

### 10.41.2 Member Function Documentation

#### 10.41.2.1 void AIRINV::FFDisutilityParser::parse (const stdair::FFDisutilityFilePath & i-FFDisutilityInputFilename, stdair::BomRoot &) [static]

Parse the CSV file describing the FFDisutility curves for the simulator, and generates the curves accordingly.

#### Parameters:

- const* stdair::Filename\_T& The file-name of the CSV-formatted FFDisutility curve input file.
- stdair::BomRoot&* Root of the BOM tree.

Definition at line 20 of file FFDisutilityParser.cpp.

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

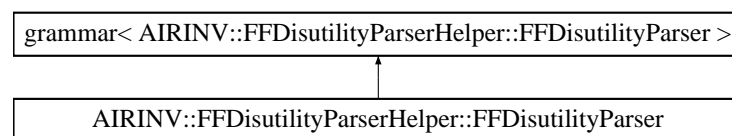
The documentation for this class was generated from the following files:

- airinv/command/[FFDisutilityParser.hpp](#)
- airinv/command/[FFDisutilityParser.cpp](#)

## 10.42 AIRINV::FFDisutilityParserHelper::FFDisutilityParser Struct Reference

```
#include <airinv/command/FFDisutilityParserHelper.hpp>
```

Inheritance diagram for AIRINV::FFDisutilityParserHelper::FFDisutilityParser::



### Public Member Functions

- [FFDisutilityParser](#) (stdair::BomRoot &, [FFDisutilityStruct](#) &)

### Public Attributes

- stdair::BomRoot & [\\_bomRoot](#)
- [FFDisutilityStruct](#) & [\\_ffDisutility](#)

### Classes

- struct [definition](#)



### 10.42.1 Detailed Description

Grammar for the FFDisutility curve parser.

Definition at line 89 of file FFDisutilityParserHelper.hpp.

### 10.42.2 Constructor & Destructor Documentation

#### 10.42.2.1 AIRINV::FFDisutilityParserHelper::FFDisutilityParser::FFDisutilityParser (stdair::BomRoot &, [FFDisutilityStruct](#) &)

Definition at line 114 of file FFDisutilityParserHelper.cpp.

### 10.42.3 Member Data Documentation

#### 10.42.3.1 stdair::BomRoot& [AIRINV::FFDisutilityParserHelper::FFDisutilityParser::\\_bomRoot](#)

Definition at line 107 of file FFDisutilityParserHelper.hpp.

Referenced by [AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::definition\(\)](#).

#### 10.42.3.2 [FFDisutilityStruct](#)& [AIRINV::FFDisutilityParserHelper::FFDisutilityParser::\\_ffDisutility](#)

Definition at line 108 of file FFDisutilityParserHelper.hpp.

Referenced by [AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::definition\(\)](#).

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

- [airinv/command/FFDisutilityParserHelper.hpp](#)
- [airinv/command/FFDisutilityParserHelper.cpp](#)

## 10.43 AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT > Struct Template Reference

```
#include <airinv/command/FFDisutilityParserHelper.hpp>
```

### Public Member Functions

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

### Public Attributes

- [boost::spirit::classic::rule< ScannerT > curve\\_list](#)
- [boost::spirit::classic::rule< ScannerT > not\\_to\\_be\\_parsed](#)
- [boost::spirit::classic::rule< ScannerT > curve](#)
- [boost::spirit::classic::rule< ScannerT > key](#)

- [boost::spirit::classic::rule< ScannerT > map](#)
- [boost::spirit::classic::rule< ScannerT > value\\_pair](#)
- [boost::spirit::classic::rule< ScannerT > curve\\_end](#)

### 10.43.1 Detailed Description

**template<typename ScannerT> struct AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >**

Definition at line 95 of file FFDisutilityParserHelper.hpp.

### 10.43.2 Constructor & Destructor Documentation

**10.43.2.1 template<typename ScannerT> [AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::definition \(FFDisutilityParser const & self\)](#)**

Definition at line 123 of file FFDisutilityParserHelper.cpp.

References [AIRINV::FFDisutilityParserHelper::FFDisutilityParser::\\_bomRoot](#), [AIRINV::FFDisutilityParserHelper::FFDisutilityParser::\\_ffDisutility](#), [AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::curve](#), [AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::curve\\_end](#), [AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::curve\\_list](#), [AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::key](#), [AIRINV::FFDisutilityParserHelper::key\\_p\(\)](#), [AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::map](#), [AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::not\\_to\\_be\\_parsed](#), and [AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::value\\_pair](#).

### 10.43.3 Member Function Documentation

**10.43.3.1 template<typename ScannerT> [bsc::rule< ScannerT > const & AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::start \(\) const](#)**

Entry point of the parser.

Definition at line 163 of file FFDisutilityParserHelper.cpp.

References [AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::curve\\_list](#).

### 10.43.4 Member Data Documentation

**10.43.4.1 template<typename ScannerT> [boost::spirit::classic::rule<ScannerT> AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::curve\\_list](#)**

Definition at line 99 of file FFDisutilityParserHelper.hpp.

Referenced by [AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::definition\(\)](#), and [AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::start\(\)](#).

**10.43.4.2 template<typename ScannerT> [boost::spirit::classic::rule<ScannerT> AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::not\\_to\\_be\\_parsed](#)**

Definition at line 99 of file FFDisutilityParserHelper.hpp.

Referenced by AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::definition().

#### 10.43.4.3 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::curve`

Definition at line 99 of file FFDisutilityParserHelper.hpp.

Referenced by AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::definition().

#### 10.43.4.4 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::key`

Definition at line 99 of file FFDisutilityParserHelper.hpp.

Referenced by AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::definition().

#### 10.43.4.5 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::map`

Definition at line 99 of file FFDisutilityParserHelper.hpp.

Referenced by AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::definition().

#### 10.43.4.6 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::value_pair`

Definition at line 99 of file FFDisutilityParserHelper.hpp.

Referenced by AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::definition().

#### 10.43.4.7 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::curve_end`

Definition at line 99 of file FFDisutilityParserHelper.hpp.

Referenced by AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >::definition().

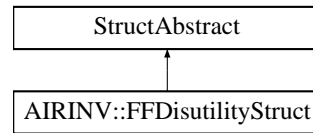
The documentation for this struct was generated from the following files:

- [airinv/command/FFDisutilityParserHelper.hpp](#)
- [airinv/command/FFDisutilityParserHelper.cpp](#)

## 10.44 AIRINV::FFDisutilityStruct Struct Reference

```
#include <airinv/bom/FFDisutilityStruct.hpp>
```

Inheritance diagram for AIRINV::FFDisutilityStruct::



### Public Member Functions

- `const std::string describe () const`
- `FFDisutilityStruct ()`
- `~FFDisutilityStruct ()`

### Public Attributes

- `std::string _key`
- `stdair::FFDisutilityCurve_T _curve`
- `stdair::DTD_T _dtd`

#### 10.44.1 Detailed Description

Utility Structure for the parsing of FFDisutility structures.

Definition at line 15 of file FFDisutilityStruct.hpp.

#### 10.44.2 Constructor & Destructor Documentation

##### 10.44.2.1 AIRINV::FFDisutilityStruct::FFDisutilityStruct ()

Default constructor.

Definition at line 15 of file FFDisutilityStruct.cpp.

##### 10.44.2.2 AIRINV::FFDisutilityStruct::~~FFDisutilityStruct ()

Destructor

Definition at line 19 of file FFDisutilityStruct.cpp.

#### 10.44.3 Member Function Documentation

##### 10.44.3.1 const std::string AIRINV::FFDisutilityStruct::describe () const

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

Definition at line 23 of file FFDisutilityStruct.cpp.

References `_curve`, and `_key`.

Referenced by `AIRINV::FFDisutilityParserHelper::doEndCurve::operator()()`.

### 10.44.4 Member Data Documentation

#### 10.44.4.1 `std::string` [AIRINV::FFDisutilityStruct::\\_key](#)

Curve key.

Definition at line 37 of file `FFDisutilityStruct.hpp`.

Referenced by `describe()`, `AIRINV::FFDisutilityParserHelper::doEndCurve::operator()()`, and `AIRINV::FFDisutilityParserHelper::storeCurveKey::operator()()`.

#### 10.44.4.2 `stdair::FFDisutilityCurve_T` [AIRINV::FFDisutilityStruct::\\_curve](#)

Curve.

Definition at line 40 of file `FFDisutilityStruct.hpp`.

Referenced by `describe()`, `AIRINV::FFDisutilityParserHelper::doEndCurve::operator()()`, and `AIRINV::FFDisutilityParserHelper::storeFFDisutilityValue::operator()()`.

#### 10.44.4.3 `stdair::DTD_T` [AIRINV::FFDisutilityStruct::\\_dtd](#)

DTD.

Definition at line 45 of file `FFDisutilityStruct.hpp`.

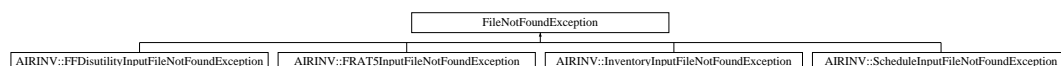
Referenced by `AIRINV::FFDisutilityParserHelper::storeFFDisutilityValue::operator()()`, and `AIRINV::FFDisutilityParserHelper::storeDTD::operator()()`.

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

- `airinv/bom/FFDisutilityStruct.hpp`
- `airinv/bom/FFDisutilityStruct.cpp`

## 10.45 FileNotFoundException Class Reference

Inheritance diagram for `FileNotFoundException`:



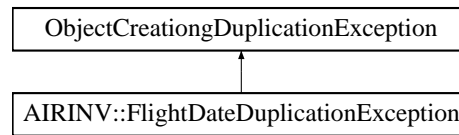
The documentation for this class was generated from the following file:

- `airinv/AIRINV_Types.hpp`

## 10.46 AIRINV::FlightDateDuplicationException Class Reference

```
#include <airinv/AIRINV_Types.hpp>
```

Inheritance diagram for `AIRINV::FlightDateDuplicationException`:



## Public Member Functions

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

### 10.46.1 Detailed Description

Duplicated flight date object.

Definition at line 154 of file AIRINV\_Types.hpp.

### 10.46.2 Constructor & Destructor Documentation

**10.46.2.1 AIRINV::FlightDateDuplicationException::FlightDateDuplicationException** (const std::string &*iWhat*) [inline]

Constructor.

Definition at line 159 of file AIRINV\_Types.hpp.

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

- [airinv/AIRINV\\_Types.hpp](#)

## 10.47 AIRINV::FlightDateHelper Class Reference

```
#include <airinv/bom/FlightDateHelper.hpp>
```

## Static Public Member Functions

- static void [fillFromRouting](#) (const stdair::FlightDate &)
- static void [updateAvailability](#) (const stdair::FlightDate &, const stdair::SegmentCabin &, const stdair::PartySize\_T &iNbOfBookings)
- static void [updateAvailabilityPool](#) (const stdair::FlightDate &, const stdair::CabinCode\_T &)
- static void [recalculateAvailability](#) (const stdair::FlightDate &, const stdair::CabinCode\_T &)
- static void [updateBookingControls](#) (stdair::FlightDate &)
- static void [recalculateAvailability](#) (const stdair::FlightDate &)

### 10.47.1 Detailed Description

Class representing the actual business functions for an airline flight-date.

Definition at line 19 of file FlightDateHelper.hpp.

## 10.47.2 Member Function Documentation

### 10.47.2.1 void AIRINV::FlightDateHelper::fillFromRouting (const stdair::FlightDate &) [static]

Fill the attributes derived from the routing legs (e.g., board and off dates).

Definition at line 51 of file FlightDateHelper.cpp.

References AIRINV::SegmentDateHelper::fillFromRouting().

Referenced by AIRINV::InventoryHelper::fillFromRouting().

### 10.47.2.2 void AIRINV::FlightDateHelper::updateAvailability (const stdair::FlightDate &, const stdair::SegmentCabin &, const stdair::PartySize\_T & iNbOfBookings) [static]

Update the availability of all classes after a reservation.

Definition at line 67 of file FlightDateHelper.cpp.

References recalculateAvailability(), and updateAvailabilityPool().

Referenced by AIRINV::SegmentCabinHelper::updateFromReservation().

### 10.47.2.3 void AIRINV::FlightDateHelper::updateAvailabilityPool (const stdair::FlightDate &, const stdair::CabinCode\_T &) [static]

Update the availability pool of all the segment-cabins after a reservation.

Definition at line 92 of file FlightDateHelper.cpp.

Referenced by updateAvailability().

### 10.47.2.4 void AIRINV::FlightDateHelper::recalculateAvailability (const stdair::FlightDate &, const stdair::CabinCode\_T &) [static]

Recalculate the availability of all the segment-cabins after a reservation.

Definition at line 127 of file FlightDateHelper.cpp.

References AIRINV::SegmentCabinHelper::updateAvailabilities().

Referenced by updateAvailability().

### 10.47.2.5 void AIRINV::FlightDateHelper::updateBookingControls (stdair::FlightDate &) [static]

Update booking controls after optimisation.

Definition at line 22 of file FlightDateHelper.cpp.

References AIRINV::SegmentCabinHelper::buildPseudoBidPriceVector(), and AIRINV::SegmentCabinHelper::updateBookingControlsUsingPseudoBidPriceVector().

### 10.47.2.6 void AIRINV::FlightDateHelper::recalculateAvailability (const stdair::FlightDate &) [static]

Recalculate the availability of all the segment-cabins after an optimisation.

Definition at line 145 of file FlightDateHelper.cpp.

References AIRINV::SegmentCabinHelper::updateAvailabilities().

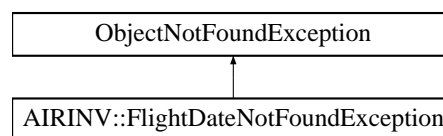
The documentation for this class was generated from the following files:

- [airinv/bom/FlightDateHelper.hpp](#)
- [airinv/bom/FlightDateHelper.cpp](#)

## 10.48 AIRINV::FlightDateNotFoundException Class Reference

```
#include <airinv/AIRINV_Types.hpp>
```

Inheritance diagram for AIRINV::FlightDateNotFoundException::



### Public Member Functions

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

#### 10.48.1 Detailed Description

Flight Date not found

Definition at line 184 of file AIRINV\_Types.hpp.

#### 10.48.2 Constructor & Destructor Documentation

**10.48.2.1 AIRINV::FlightDateNotFoundException::FlightDateNotFoundException** (const std::string &*iWhat*) [inline]

Constructor.

Definition at line 189 of file AIRINV\_Types.hpp.

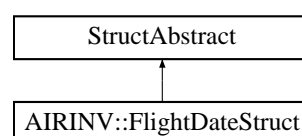
The documentation for this class was generated from the following file:

- [airinv/AIRINV\\_Types.hpp](#)

## 10.49 AIRINV::FlightDateStruct Struct Reference

```
#include <airinv/bom/FlightDateStruct.hpp>
```

Inheritance diagram for AIRINV::FlightDateStruct::





### 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](#) &)
- [FlightDateStruct](#) ()

### Public Attributes

- stdair::AirlineCode\_T [\\_airlineCode](#)
- stdair::FlightNumber\_T [\\_flightNumber](#)
- stdair::Date\_T [\\_flightDate](#)
- [FlightTypeCode](#) [\\_flightTypeCode](#)
- [FlightVisibilityCode](#) [\\_flightVisibilityCode](#)
- [LegStructList\\_T](#) [\\_legList](#)
- [SegmentStructList\\_T](#) [\\_segmentList](#)
- unsigned int [\\_itYear](#)
- unsigned int [\\_itMonth](#)
- unsigned int [\\_itDay](#)
- int [\\_dateOffSet](#)
- long [\\_itHours](#)
- long [\\_itMinutes](#)
- long [\\_itSeconds](#)
- [AirportList\\_T](#) [\\_airportList](#)
- [AirportOrderedList\\_T](#) [\\_airportOrderedList](#)
- bool [\\_legAlreadyDefined](#)
- [LegStruct](#) [\\_itLeg](#)
- [LegCabinStruct](#) [\\_itLegCabin](#)
- [BucketStruct](#) [\\_itBucket](#)
- bool [\\_areSegmentDefinitionsSpecific](#)
- [SegmentStruct](#) [\\_itSegment](#)
- [SegmentCabinStruct](#) [\\_itSegmentCabin](#)
- [BookingClassStruct](#) [\\_itBookingClass](#)

#### 10.49.1 Detailed Description

Utility Structure for the parsing of Flight-Date structures.

Definition at line 27 of file [FlightDateStruct.hpp](#).

## 10.49.2 Constructor & Destructor Documentation

### 10.49.2.1 AIRINV::FlightDateStruct::FlightDateStruct ()

Constructor.

Definition at line 17 of file FlightDateStruct.cpp.

## 10.49.3 Member Function Documentation

### 10.49.3.1 stdair::Date\_T AIRINV::FlightDateStruct::getDate () const

Set the date from the staging details.

Definition at line 25 of file FlightDateStruct.cpp.

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

Referenced by `AIRINV::InventoryParserHelper::storeOffDate::operator()`, `AIRINV::InventoryParserHelper::storeBoardingDate::operator()`, `AIRINV::InventoryParserHelper::storeFlightDate::operator()`, and `AIRINV::InventoryParserHelper::storeSnapshotDate::operator()`.

### 10.49.3.2 stdair::Duration\_T AIRINV::FlightDateStruct::getTime () const

Set the time from the staging details.

Definition at line 30 of file FlightDateStruct.cpp.

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

Referenced by `AIRINV::InventoryParserHelper::storeOffTime::operator()`, and `AIRINV::InventoryParserHelper::storeBoardingTime::operator()`.

### 10.49.3.3 const std::string AIRINV::FlightDateStruct::describe () const

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

Definition at line 37 of file FlightDateStruct.cpp.

References `_airlineCode`, `_flightDate`, `_flightNumber`, `_flightTypeCode`, `_flightVisibilityCode`, `_legList`, `_segmentList`, `AIRINV::FlightVisibilityCode::describe()`, `AIRINV::FlightVisibilityCode::getCode()`, and `AIRINV::FlightVisibilityCode::NORMAL`.

### 10.49.3.4 void AIRINV::FlightDateStruct::addAirport (const stdair::AirportCode\_T &)

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

Definition at line 67 of file FlightDateStruct.cpp.

References `_airportList`, and `_airportOrderedList`.

Referenced by `AIRINV::InventoryParserHelper::storeLegOffPoint::operator()`, and `AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator()`.

### 10.49.3.5 void AIRINV::FlightDateStruct::buildSegments ()

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

Definition at line 83 of file FlightDateStruct.cpp.

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

**10.49.3.6 void AIRINV::FlightDateStruct::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 116 of file FlightDateStruct.cpp.

References AIRINV::SegmentStruct::\_boardingPoint, AIRINV::SegmentStruct::\_offPoint, and \_segmentList.

**10.49.3.7 void AIRINV::FlightDateStruct::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 153 of file FlightDateStruct.cpp.

References \_segmentList.

**10.49.3.8 void AIRINV::FlightDateStruct::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 167 of file FlightDateStruct.cpp.

References AIRINV::SegmentStruct::\_boardingPoint, AIRINV::SegmentCabinStruct::\_cabinCode, AIRINV::SegmentStruct::\_offPoint, and \_segmentList.

**10.49.3.9 void AIRINV::FlightDateStruct::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 231 of file FlightDateStruct.cpp.

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

**10.49.4 Member Data Documentation****10.49.4.1 stdair::AirlineCode\_T AIRINV::FlightDateStruct::\_airlineCode**

Definition at line 81 of file FlightDateStruct.hpp.

Referenced by describe(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator>(), and AIRINV::InventoryParserHelper::storeAirlineCode::operator().

**10.49.4.2 stdair::FlightNumber\_T AIRINV::FlightDateStruct::\_flightNumber**

Definition at line 82 of file FlightDateStruct.hpp.

Referenced by describe(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), and AIRINV::InventoryParserHelper::storeFlightNumber::operator().

**10.49.4.3 stdair::Date\_T AIRINV::FlightDateStruct::\_flightDate**

Definition at line 83 of file FlightDateStruct.hpp.

Referenced by describe(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

**10.49.4.4 FlightTypeCode AIRINV::FlightDateStruct::\_flightTypeCode**

Definition at line 84 of file FlightDateStruct.hpp.

Referenced by describe(), and AIRINV::InventoryParserHelper::storeFlightTypeCode::operator().

**10.49.4.5 FlightVisibilityCode AIRINV::FlightDateStruct::\_flightVisibilityCode**

Definition at line 85 of file FlightDateStruct.hpp.

Referenced by describe(), and AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator().

**10.49.4.6 LegStructList\_T AIRINV::FlightDateStruct::\_legList**

Definition at line 86 of file FlightDateStruct.hpp.

Referenced by describe(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), and AIRINV::InventoryParserHelper::storeAirlineCode::operator().

**10.49.4.7 SegmentStructList\_T AIRINV::FlightDateStruct::\_segmentList**

Definition at line 87 of file FlightDateStruct.hpp.

Referenced by addFareFamily(), addSegmentCabin(), buildSegments(), describe(), AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), and AIRINV::InventoryParserHelper::storeAirlineCode::operator().

**10.49.4.8 unsigned int AIRINV::FlightDateStruct::\_itYear**

Staging Date.

Definition at line 90 of file FlightDateStruct.hpp.

Referenced by getDate().

**10.49.4.9 unsigned int AIRINV::FlightDateStruct::\_itMonth**

Definition at line 91 of file FlightDateStruct.hpp.

Referenced by getDate().

**10.49.4.10 unsigned int AIRINV::FlightDateStruct::\_itDay**

Definition at line 92 of file FlightDateStruct.hpp.

Referenced by getDate().

**10.49.4.11 int AIRINV::FlightDateStruct::\_dateOffset**

Definition at line 93 of file FlightDateStruct.hpp.

Referenced by AIRINV::InventoryParserHelper::storeBoardingTime::operator()().

**10.49.4.12 long AIRINV::FlightDateStruct::\_itHours**

Staging Time.

Definition at line 96 of file FlightDateStruct.hpp.

Referenced by getTime().

**10.49.4.13 long AIRINV::FlightDateStruct::\_itMinutes**

Definition at line 97 of file FlightDateStruct.hpp.

Referenced by getTime().

**10.49.4.14 long AIRINV::FlightDateStruct::\_itSeconds**

Definition at line 98 of file FlightDateStruct.hpp.

Referenced by getTime(), AIRINV::InventoryParserHelper::storeOffTime::operator()(), and AIRINV::InventoryParserHelper::storeBoardingTime::operator()().

**10.49.4.15 AirportList\_T AIRINV::FlightDateStruct::\_airportList**

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

Definition at line 102 of file FlightDateStruct.hpp.

Referenced by addAirport(), and buildSegments().

**10.49.4.16 AirportOrderedList\_T AIRINV::FlightDateStruct::\_airportOrderedList**

Definition at line 103 of file FlightDateStruct.hpp.

Referenced by addAirport(), and buildSegments().

**10.49.4.17 bool AIRINV::FlightDateStruct::\_legAlreadyDefined**

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

Definition at line 107 of file FlightDateStruct.hpp.

**10.49.4.18 LegStruct AIRINV::FlightDateStruct::\_itLeg**

Definition at line 108 of file FlightDateStruct.hpp.

Referenced by AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), and AIRINV::InventoryParserHelper::storeAirlineCode::operator().

#### 10.49.4.19 LegCabinStruct AIRINV::FlightDateStruct::\_itLegCabin

Definition at line 109 of file FlightDateStruct.hpp.

Referenced by AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), and AIRINV::InventoryParserHelper::storeAirlineCode::operator().

#### 10.49.4.20 BucketStruct AIRINV::FlightDateStruct::\_itBucket

Definition at line 110 of file FlightDateStruct.hpp.

Referenced by AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), and AIRINV::InventoryParserHelper::storeAirlineCode::operator().

#### 10.49.4.21 bool AIRINV::FlightDateStruct::\_areSegmentDefinitionsSpecific

Staging Segment-related attributes.

Definition at line 113 of file FlightDateStruct.hpp.

#### 10.49.4.22 SegmentStruct AIRINV::FlightDateStruct::\_itSegment

Definition at line 114 of file FlightDateStruct.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), and AIRINV::InventoryParserHelper::storeAirlineCode::operator().

#### 10.49.4.23 SegmentCabinStruct AIRINV::FlightDateStruct::\_itSegmentCabin

Definition at line 115 of file FlightDateStruct.hpp.

Referenced by AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), and AIRINV::InventoryParserHelper::storeAirlineCode::operator().

#### 10.49.4.24 BookingClassStruct AIRINV::FlightDateStruct::\_itBookingClass

Definition at line 116 of file FlightDateStruct.hpp.

Referenced by AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), and AIRINV::InventoryParserHelper::storeAirlineCode::operator().

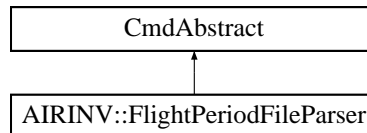
The documentation for this struct was generated from the following files:

- [airinv/bom/FlightDateStruct.hpp](#)
- [airinv/bom/FlightDateStruct.cpp](#)

## 10.50 AIRINV::FlightPeriodFileParser Class Reference

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

Inheritance diagram for AIRINV::FlightPeriodFileParser:



### Public Member Functions

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

#### 10.50.1 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 325 of file ScheduleParserHelper.hpp.

### 10.50.2 Constructor & Destructor Documentation

#### 10.50.2.1 AIRINV::FlightPeriodFileParser::FlightPeriodFileParser (stdair::BomRoot & ioBomRoot, const stdair::Filename\_T & iFilename)

Constructor.

Definition at line 706 of file ScheduleParserHelper.cpp.

### 10.50.3 Member Function Documentation

#### 10.50.3.1 bool AIRINV::FlightPeriodFileParser::generateInventories ()

Parse the input file and generate the Inventories.

Definition at line 730 of file ScheduleParserHelper.cpp.

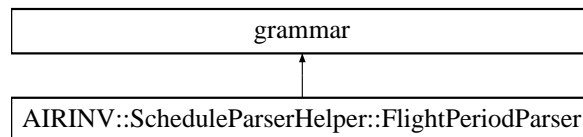
The documentation for this class was generated from the following files:

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

## 10.51 AIRINV::ScheduleParserHelper::FlightPeriodParser Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::FlightPeriodParser::



### Public Member Functions

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

### Public Attributes

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

### Classes

- struct [definition](#)



### 10.51.1 Detailed Description

Grammar for the Flight-Period parser.

Definition at line 282 of file ScheduleParserHelper.hpp.

### 10.51.2 Constructor & Destructor Documentation

#### 10.51.2.1 AIRINV::ScheduleParserHelper::FlightPeriodParser::FlightPeriodParser (stdair::BomRoot &, [FlightPeriodStruct](#) &)

Definition at line 529 of file ScheduleParserHelper.cpp.

### 10.51.3 Member Data Documentation

#### 10.51.3.1 stdair::BomRoot& [AIRINV::ScheduleParserHelper::FlightPeriodParser::\\_bomRoot](#)

Definition at line 306 of file ScheduleParserHelper.hpp.

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

#### 10.51.3.2 [FlightPeriodStruct&](#) [AIRINV::ScheduleParserHelper::FlightPeriodParser::\\_flight-Period](#)

Definition at line 307 of file ScheduleParserHelper.hpp.

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

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

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

## 10.52 AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT > Struct Template Reference

```
#include <airinv/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 > not\\_to\\_be\\_parsed](#)
- [boost::spirit::classic::rule< ScannerT > flight\\_period](#)
- [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](#)

### 10.52.1 Detailed Description

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

Definition at line 288 of file ScheduleParserHelper.hpp.

### 10.52.2 Constructor & Destructor Documentation

**10.52.2.1 template<typename ScannerT> AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::definition (FlightPeriodParser const & self)**

Definition at line 538 of file ScheduleParserHelper.cpp.

References AIRINV::ScheduleParserHelper::FlightPeriodParser::\_bomRoot, AIRINV::ScheduleParserHelper::FlightPeriodParser::\_flightPeriod, AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::airline\_code, AIRINV::ScheduleParserHelper::airline\_code\_p(), AIRINV::ScheduleParserHelper::airport\_p(), AIRINV::ScheduleParserHelper::cabin\_code\_p(), AIRINV::ScheduleParserHelper::class\_code\_list\_p(), AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::date, AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::date\_offset, AIRINV::ScheduleParserHelper::day\_p(), AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::dow, AIRINV::ScheduleParserHelper::dow\_p(), AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::family\_cabin\_details, AIRINV::ScheduleParserHelper::family\_code\_p, AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::flight\_key, AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::flight\_number, AIRINV::ScheduleParserHelper::flight\_number\_p(), AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::flight\_period, AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::flight\_period\_end, AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::flight\_period\_list, AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::full\_segment\_cabin\_details, AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::generic\_segment,

AIRINV::ScheduleParserHelper::hours\_p(), AIRINV::ScheduleParserHelper::int1\_p(), AIRINV::ScheduleParserHelper::key\_p(), AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::leg, AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::leg\_cabin\_details, AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::leg\_details, AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::leg\_key, AIRINV::ScheduleParserHelper::minutes\_p(), AIRINV::ScheduleParserHelper::month\_p(), AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::not\_to\_be\_parsed, AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::operating\_leg\_details, AIRINV::ScheduleParserHelper::seconds\_p(), AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::segment\_cabin\_details, AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::segment\_key, AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::segment\_section, AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::specific\_segment\_list, AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::time, and AIRINV::ScheduleParserHelper::year\_p().

### 10.52.3 Member Function Documentation

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

Entry point of the parser.

Definition at line 692 of file ScheduleParserHelper.cpp.

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

### 10.52.4 Member Data Documentation

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

**10.52.4.4** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::flight_period_end`

Definition at line 292 of file ScheduleParserHelper.hpp.

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

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

**10.52.4.14** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::operating_leg_details`

Definition at line 292 of file ScheduleParserHelper.hpp.

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

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

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

Definition at line 292 of file ScheduleParserHelper.hpp.

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

#### 10.52.4.24 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >::specific_segment_list`

Definition at line 292 of file `ScheduleParserHelper.hpp`.

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

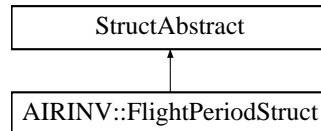
The documentation for this struct was generated from the following files:

- `airinv/command/ScheduleParserHelper.hpp`
- `airinv/command/ScheduleParserHelper.cpp`

### 10.53 AIRINV::FlightPeriodStruct Struct Reference

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

Inheritance diagram for AIRINV::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](#)

### 10.53.1 Detailed Description

Utility Structure for the parsing of Flight-Period structures.

Definition at line 24 of file FlightPeriodStruct.hpp.

### 10.53.2 Constructor & Destructor Documentation

#### 10.53.2.1 AIRINV::FlightPeriodStruct::FlightPeriodStruct ()

Constructor.

Definition at line 17 of file FlightPeriodStruct.cpp.

### 10.53.3 Member Function Documentation

#### 10.53.3.1 stdair::Date\_T AIRINV::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 [AIRINV::ScheduleParserHelper::storeDateRangeEnd::operator\(\)\(\)](#), and [AIRINV::ScheduleParserHelper::storeDateRangeStart::operator\(\)\(\)](#).

#### 10.53.3.2 stdair::Duration\_T AIRINV::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 [AIRINV::ScheduleParserHelper::storeElapsedTime::operator\(\)\(\)](#), [AIRINV::ScheduleParserHelper::storeOffTime::operator\(\)\(\)](#), and [AIRINV::ScheduleParserHelper::storeBoardingTime::operator\(\)\(\)](#).



**10.53.3.3 const std::string AIRINV::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 `AIRINV::ScheduleParserHelper::doEndFlight::operator()()`.

**10.53.3.4 void AIRINV::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 `AIRINV::ScheduleParserHelper::storeLegOffPoint::operator()()`, and `AIRINV::ScheduleParserHelper::storeLegBoardingPoint::operator()()`.

**10.53.3.5 void AIRINV::FlightPeriodStruct::buildSegments ()**

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

Definition at line 78 of file FlightPeriodStruct.cpp.

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

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

**10.53.3.6 void AIRINV::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 `AIRINV::SegmentStruct::_boardingPoint`, `AIRINV::SegmentStruct::_offPoint`, and `_segmentList`.

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

**10.53.3.7 void AIRINV::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 148 of file FlightPeriodStruct.cpp.

References `_segmentList`.

**10.53.3.8 void AIRINV::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 161 of file FlightPeriodStruct.cpp.

References AIRINV::SegmentStruct::\_boardingPoint, AIRINV::SegmentCabinStruct::\_cabinCode, AIRINV::SegmentStruct::\_offPoint, and \_segmentList.

Referenced by AIRINV::ScheduleParserHelper::storeFClasses::operator()().

#### 10.53.3.9 void AIRINV::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 225 of file FlightPeriodStruct.cpp.

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

### 10.53.4 Member Data Documentation

#### 10.53.4.1 stdair::AirlineCode\_T AIRINV::FlightPeriodStruct::\_airlineCode

Definition at line 80 of file FlightPeriodStruct.hpp.

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

#### 10.53.4.2 stdair::FlightNumber\_T AIRINV::FlightPeriodStruct::\_flightNumber

Definition at line 81 of file FlightPeriodStruct.hpp.

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

#### 10.53.4.3 stdair::DatePeriod\_T AIRINV::FlightPeriodStruct::\_dateRange

Definition at line 82 of file FlightPeriodStruct.hpp.

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

#### 10.53.4.4 stdair::DoWStruct AIRINV::FlightPeriodStruct::\_dow

Definition at line 83 of file FlightPeriodStruct.hpp.

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

#### 10.53.4.5 LegStructList\_T AIRINV::FlightPeriodStruct::\_legList

Definition at line 84 of file FlightPeriodStruct.hpp.

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

#### 10.53.4.6 SegmentStructList\_T AIRINV::FlightPeriodStruct::\_segmentList

Definition at line 85 of file FlightPeriodStruct.hpp.

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

#### 10.53.4.7 bool AIRINV::FlightPeriodStruct::\_legAlreadyDefined

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

Definition at line 89 of file FlightPeriodStruct.hpp.

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

#### 10.53.4.8 LegStruct AIRINV::FlightPeriodStruct::\_itLeg

Definition at line 90 of file FlightPeriodStruct.hpp.

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

#### 10.53.4.9 LegCabinStruct AIRINV::FlightPeriodStruct::\_itLegCabin

Definition at line 91 of file FlightPeriodStruct.hpp.

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

#### 10.53.4.10 std::date\_T AIRINV::FlightPeriodStruct::\_dateRangeStart

Staging Date.

Definition at line 94 of file FlightPeriodStruct.hpp.

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

#### 10.53.4.11 std::date\_T AIRINV::FlightPeriodStruct::\_dateRangeEnd

Definition at line 95 of file FlightPeriodStruct.hpp.

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

#### 10.53.4.12 unsigned int AIRINV::FlightPeriodStruct::\_itYear

Definition at line 96 of file FlightPeriodStruct.hpp.

Referenced by getDate().

**10.53.4.13 unsigned int AIRINV::FlightPeriodStruct::\_itMonth**

Definition at line 97 of file FlightPeriodStruct.hpp.

Referenced by getDate().

**10.53.4.14 unsigned int AIRINV::FlightPeriodStruct::\_itDay**

Definition at line 98 of file FlightPeriodStruct.hpp.

Referenced by getDate().

**10.53.4.15 int AIRINV::FlightPeriodStruct::\_dateOffset**

Definition at line 99 of file FlightPeriodStruct.hpp.

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

**10.53.4.16 long AIRINV::FlightPeriodStruct::\_itHours**

Staging Time.

Definition at line 102 of file FlightPeriodStruct.hpp.

Referenced by getTime().

**10.53.4.17 long AIRINV::FlightPeriodStruct::\_itMinutes**

Definition at line 103 of file FlightPeriodStruct.hpp.

Referenced by getTime().

**10.53.4.18 long AIRINV::FlightPeriodStruct::\_itSeconds**

Definition at line 104 of file FlightPeriodStruct.hpp.

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

**10.53.4.19 AirportList\_T AIRINV::FlightPeriodStruct::\_airportList**

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

Definition at line 108 of file FlightPeriodStruct.hpp.

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

**10.53.4.20 AirportOrderedList\_T AIRINV::FlightPeriodStruct::\_airportOrderedList**

Definition at line 109 of file FlightPeriodStruct.hpp.

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

**10.53.4.21 bool AIRINV::FlightPeriodStruct::\_areSegmentDefinitionsSpecific**

Staging Segment-related attributes.

Definition at line 112 of file FlightPeriodStruct.hpp.

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

**10.53.4.22 SegmentStruct AIRINV::FlightPeriodStruct::\_itSegment**

Definition at line 113 of file FlightPeriodStruct.hpp.

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

**10.53.4.23 SegmentCabinStruct AIRINV::FlightPeriodStruct::\_itSegmentCabin**

Definition at line 114 of file FlightPeriodStruct.hpp.

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

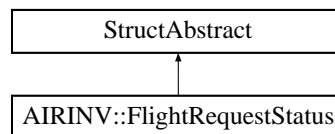
The documentation for this struct was generated from the following files:

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

**10.54 AIRINV::FlightRequestStatus Struct Reference**

```
#include <airinv/FlightRequestStatus.hpp>
```

Inheritance diagram for AIRINV::FlightRequestStatus::

**Public Types**

- `OK = 0`
- `NOT_FOUND`
- `INTERNAL_ERROR`
- `LAST_VALUE`
- `enum EN_FlightRequestStatus { OK = 0, NOT_FOUND, INTERNAL_ERROR, LAST_VALUE }`

### Public Member Functions

- [EN\\_FlightRequestStatus getCode](#) () const
- const std::string [describe](#) () const
- [FlightRequestStatus](#) (const [EN\\_FlightRequestStatus](#) &)
- [FlightRequestStatus](#) (const std::string &iCode)

### Static Public Member Functions

- static const std::string & [getLabel](#) (const [EN\\_FlightRequestStatus](#) &)
- static const std::string & [getCodeLabel](#) (const [EN\\_FlightRequestStatus](#) &)
- static std::string [describeLabels](#) ()

#### 10.54.1 Detailed Description

Enumeration of flight type codes.

Definition at line 15 of file FlightRequestStatus.hpp.

#### 10.54.2 Member Enumeration Documentation

##### 10.54.2.1 enum [AIRINV::FlightRequestStatus::EN\\_FlightRequestStatus](#)

Enumerator:

*OK*  
*NOT\_FOUND*  
*INTERNAL\_ERROR*  
*LAST\_VALUE*

Definition at line 17 of file FlightRequestStatus.hpp.

#### 10.54.3 Constructor & Destructor Documentation

##### 10.54.3.1 [AIRINV::FlightRequestStatus::FlightRequestStatus](#) (const [EN\\_FlightRequestStatus](#) &)

Constructor.

Definition at line 25 of file FlightRequestStatus.cpp.

##### 10.54.3.2 [AIRINV::FlightRequestStatus::FlightRequestStatus](#) (const std::string &iCode)

Constructor.

Definition at line 30 of file FlightRequestStatus.cpp.

References [describeLabels\(\)](#), [INTERNAL\\_ERROR](#), [LAST\\_VALUE](#), [NOT\\_FOUND](#), and [OK](#).

#### 10.54.4 Member Function Documentation

##### 10.54.4.1 `const std::string & AIRINV::FlightRequestStatus::getLabel (const EN_FlightRequestStatus &) [static]`

Get the label as a string.

Definition at line 58 of file FlightRequestStatus.cpp.

##### 10.54.4.2 `const std::string & AIRINV::FlightRequestStatus::getCodeLabel (const EN_FlightRequestStatus &) [static]`

Get the label as a single char.

Definition at line 64 of file FlightRequestStatus.cpp.

##### 10.54.4.3 `std::string AIRINV::FlightRequestStatus::describeLabels () [static]`

List the labels.

Definition at line 69 of file FlightRequestStatus.cpp.

References LAST\_VALUE.

Referenced by FlightRequestStatus().

##### 10.54.4.4 `FlightRequestStatus::EN_FlightRequestStatus AIRINV::FlightRequestStatus::getCode () const`

Get the enumerated value.

Definition at line 82 of file FlightRequestStatus.cpp.

##### 10.54.4.5 `const std::string AIRINV::FlightRequestStatus::describe () const`

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

Definition at line 87 of file FlightRequestStatus.cpp.

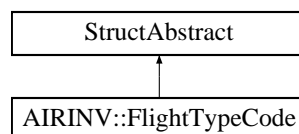
The documentation for this struct was generated from the following files:

- [airinv/FlightRequestStatus.hpp](#)
- [airinv/basic/FlightRequestStatus.cpp](#)

## 10.55 AIRINV::FlightTypeCode Struct Reference

```
#include <airinv/basic/FlightTypeCode.hpp>
```

Inheritance diagram for AIRINV::FlightTypeCode::



## Public Types

- `DOMESTIC` = 0
- `INTERNATIONAL`
- `GROUND_HANDLING`
- `LAST_VALUE`
- enum `EN_FlightTypeCode` { `DOMESTIC` = 0, `INTERNATIONAL`, `GROUND_HANDLING`, `LAST_VALUE` }

## Public Member Functions

- `EN_FlightTypeCode` `getCode` () const
- const std::string `describe` () const
- `FlightTypeCode` (const `EN_FlightTypeCode` &)
- `FlightTypeCode` (const std::string &iCode)

## Static Public Member Functions

- static const std::string & `getLabel` (const `EN_FlightTypeCode` &)
- static const std::string & `getCodeLabel` (const `EN_FlightTypeCode` &)
- static std::string `describeLabels` ()

### 10.55.1 Detailed Description

Enumeration of flight type codes.

Definition at line 15 of file `FlightTypeCode.hpp`.

### 10.55.2 Member Enumeration Documentation

#### 10.55.2.1 enum `AIRINV::FlightTypeCode::EN_FlightTypeCode`

Enumerator:

*`DOMESTIC`*  
*`INTERNATIONAL`*  
*`GROUND_HANDLING`*  
*`LAST_VALUE`*

Definition at line 17 of file `FlightTypeCode.hpp`.

### 10.55.3 Constructor & Destructor Documentation

#### 10.55.3.1 `AIRINV::FlightTypeCode::FlightTypeCode` (const `EN_FlightTypeCode` &)

Constructor.

Definition at line 24 of file `FlightTypeCode.cpp`.



### 10.55.3.2 AIRINV::FlightTypeCode::FlightTypeCode (const std::string & iCode)

Constructor.

Definition at line 29 of file FlightTypeCode.cpp.

References describeLabels(), DOMESTIC, GROUND\_HANDLING, INTERNATIONAL, and LAST\_VALUE.

## 10.55.4 Member Function Documentation

### 10.55.4.1 const std::string & AIRINV::FlightTypeCode::getLabel (const EN\_FlightTypeCode &) [static]

Get the label as a string.

Definition at line 54 of file FlightTypeCode.cpp.

### 10.55.4.2 const std::string & AIRINV::FlightTypeCode::getCodeLabel (const EN\_FlightTypeCode &) [static]

Get the label as a single char.

Definition at line 60 of file FlightTypeCode.cpp.

### 10.55.4.3 std::string AIRINV::FlightTypeCode::describeLabels () [static]

List the labels.

Definition at line 65 of file FlightTypeCode.cpp.

References LAST\_VALUE.

Referenced by FlightTypeCode().

### 10.55.4.4 EN\_FlightTypeCode AIRINV::FlightTypeCode::getCode () const

Get the enumerated value.

Definition at line 77 of file FlightTypeCode.cpp.

Referenced by AIRINV::InventoryParserHelper::storeFlightTypeCode::operator()().

### 10.55.4.5 const std::string AIRINV::FlightTypeCode::describe () const

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

Definition at line 82 of file FlightTypeCode.cpp.

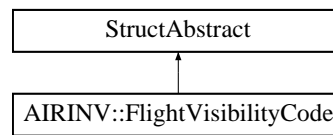
The documentation for this struct was generated from the following files:

- [airinv/basic/FlightTypeCode.hpp](#)
- [airinv/basic/FlightTypeCode.cpp](#)

## 10.56 AIRINV::FlightVisibilityCode Struct Reference

```
#include <airinv/basic/FlightVisibilityCode.hpp>
```

Inheritance diagram for AIRINV::FlightVisibilityCode::



### Public Types

- `NORMAL` = 0
- `HIDDEN`
- `PSEUDO`
- `LAST_VALUE`
- enum `EN_FlightVisibilityCode` { `NORMAL` = 0, `HIDDEN`, `PSEUDO`, `LAST_VALUE` }

### Public Member Functions

- `EN_FlightVisibilityCode` `getCode` () const
- const std::string `describe` () const
- `FlightVisibilityCode` (const `EN_FlightVisibilityCode` &)
- `FlightVisibilityCode` (const std::string &iCode)

### Static Public Member Functions

- static const std::string & `getLabel` (const `EN_FlightVisibilityCode` &)
- static const std::string & `getCodeLabel` (const `EN_FlightVisibilityCode` &)
- static std::string `describeLabels` ()

#### 10.56.1 Detailed Description

Enumeration of flight visibility codes.

Definition at line 15 of file `FlightVisibilityCode.hpp`.

#### 10.56.2 Member Enumeration Documentation

##### 10.56.2.1 enum `AIRINV::FlightVisibilityCode::EN_FlightVisibilityCode`

Enumerator:

*`NORMAL`*

*`HIDDEN`*

*`PSEUDO`*

*`LAST_VALUE`*

Definition at line 17 of file `FlightVisibilityCode.hpp`.

### 10.56.3 Constructor & Destructor Documentation

#### 10.56.3.1 AIRINV::FlightVisibilityCode::FlightVisibilityCode (const EN\_FlightVisibilityCode &)

Constructor.

Definition at line 25 of file FlightVisibilityCode.cpp.

#### 10.56.3.2 AIRINV::FlightVisibilityCode::FlightVisibilityCode (const std::string & iCode)

Constructor.

Definition at line 30 of file FlightVisibilityCode.cpp.

References describeLabels(), HIDDEN, LAST\_VALUE, NORMAL, and PSEUDO.

### 10.56.4 Member Function Documentation

#### 10.56.4.1 const std::string & AIRINV::FlightVisibilityCode::getLabel (const EN\_FlightVisibilityCode &) [static]

Get the label as a string.

Definition at line 57 of file FlightVisibilityCode.cpp.

#### 10.56.4.2 const std::string & AIRINV::FlightVisibilityCode::getCodeLabel (const EN\_FlightVisibilityCode &) [static]

Get the label as a single char.

Definition at line 63 of file FlightVisibilityCode.cpp.

#### 10.56.4.3 std::string AIRINV::FlightVisibilityCode::describeLabels () [static]

List the labels.

Definition at line 68 of file FlightVisibilityCode.cpp.

References LAST\_VALUE.

Referenced by FlightVisibilityCode().

#### 10.56.4.4 FlightVisibilityCode::EN\_FlightVisibilityCode AIRINV::FlightVisibilityCode::getCode () const

Get the enumerated value.

Definition at line 81 of file FlightVisibilityCode.cpp.

Referenced by AIRINV::FlightDateStruct::describe(), and AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator()().

#### 10.56.4.5 const std::string AIRINV::FlightVisibilityCode::describe () const

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

Definition at line 86 of file FlightVisibilityCode.cpp.

Referenced by AIRINV::FlightDateStruct::describe().

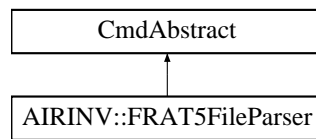
The documentation for this struct was generated from the following files:

- [airinv/basic/FlightVisibilityCode.hpp](#)
- [airinv/basic/FlightVisibilityCode.cpp](#)

## 10.57 AIRINV::FRAT5FileParser Class Reference

```
#include <airinv/command/FRAT5ParserHelper.hpp>
```

Inheritance diagram for AIRINV::FRAT5FileParser::



### Public Member Functions

- [FRAT5FileParser](#) (stdair::BomRoot &ioBomRoot, const stdair::Filename\_T &iFilename)
- bool [generateFRAT5Curves](#) ()

#### 10.57.1 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 126 of file FRAT5ParserHelper.hpp.

#### 10.57.2 Constructor & Destructor Documentation

##### 10.57.2.1 AIRINV::FRAT5FileParser::FRAT5FileParser (stdair::BomRoot & ioBomRoot, const stdair::Filename\_T & iFilename)

Constructor.

Definition at line 178 of file FRAT5ParserHelper.cpp.

#### 10.57.3 Member Function Documentation

##### 10.57.3.1 bool AIRINV::FRAT5FileParser::generateFRAT5Curves ()

Parse the input file and generate the FRAT5 curves.

Definition at line 202 of file FRAT5ParserHelper.cpp.

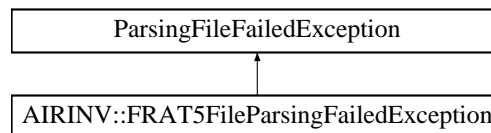
The documentation for this class was generated from the following files:

- [airinv/command/FRAT5ParserHelper.hpp](#)
- [airinv/command/FRAT5ParserHelper.cpp](#)

## 10.58 AIRINV::FRAT5FileParsingFailedException Class Reference

```
#include <airinv/AIRINV_Types.hpp>
```

Inheritance diagram for AIRINV::FRAT5FileParsingFailedException::



### Public Member Functions

- [FRAT5FileParsingFailedException](#) (const std::string &iWhat)

### 10.58.1 Detailed Description

The FRAT5 input file can not be parsed.

Definition at line 67 of file AIRINV\_Types.hpp.

### 10.58.2 Constructor & Destructor Documentation

#### 10.58.2.1 AIRINV::FRAT5FileParsingFailedException::FRAT5FileParsingFailedException (const std::string &iWhat) [inline]

Constructor.

Definition at line 73 of file AIRINV\_Types.hpp.

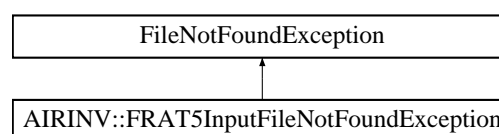
The documentation for this class was generated from the following file:

- [airinv/AIRINV\\_Types.hpp](#)

## 10.59 AIRINV::FRAT5InputFileNotFoundedException Class Reference

```
#include <airinv/AIRINV_Types.hpp>
```

Inheritance diagram for AIRINV::FRAT5InputFileNotFoundedException::



### Public Member Functions

- [FRAT5InputFileNotFoundedException](#) (const std::string &iWhat)

### 10.59.1 Detailed Description

The FRAT5 input file can not be found or opened.

Definition at line 130 of file AIRINV\_Types.hpp.

### 10.59.2 Constructor & Destructor Documentation

#### 10.59.2.1 AIRINV::FRAT5InputFileNotFoundException::FRAT5InputFileNotFoundException (const std::string & *iWhat*) [inline]

Constructor.

Definition at line 135 of file AIRINV\_Types.hpp.

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

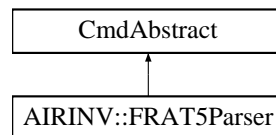
- [airinv/AIRINV\\_Types.hpp](#)

## 10.60 AIRINV::FRAT5Parser Class Reference

Class wrapping the parser entry point.

```
#include <airinv/command/FRAT5Parser.hpp>
```

Inheritance diagram for AIRINV::FRAT5Parser::



### Static Public Member Functions

- static void [parse](#) (const stdair::FRAT5FilePath &iFRAT5InputFilename, stdair::BomRoot &)

### 10.60.1 Detailed Description

Class wrapping the parser entry point.

Definition at line 22 of file FRAT5Parser.hpp.

### 10.60.2 Member Function Documentation

#### 10.60.2.1 void AIRINV::FRAT5Parser::parse (const stdair::FRAT5FilePath & *iFRAT5InputFilename*, stdair::BomRoot &) [static]

Parse the CSV file describing the FRAT5 curves for the simulator, and generates the curves accordingly.

#### Parameters:

**const** stdair::Filename\_T& The file-name of the CSV-formatted FRAT5 curve input file.

*stdair::BomRoot&* Root of the BOM tree.

Definition at line 20 of file FRAT5Parser.cpp.

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

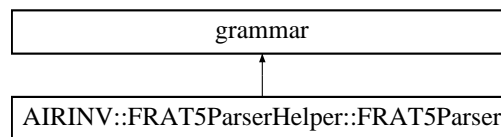
The documentation for this class was generated from the following files:

- [airinv/command/FRAT5Parser.hpp](#)
- [airinv/command/FRAT5Parser.cpp](#)

## 10.61 AIRINV::FRAT5ParserHelper::FRAT5Parser Struct Reference

```
#include <airinv/command/FRAT5ParserHelper.hpp>
```

Inheritance diagram for AIRINV::FRAT5ParserHelper::FRAT5Parser::



### Public Member Functions

- [FRAT5Parser](#) (stdair::BomRoot &, [FRAT5Struct](#) &)

### Public Attributes

- stdair::BomRoot & [\\_bomRoot](#)
- [FRAT5Struct](#) & [\\_frat5](#)

### Classes

- struct [definition](#)

#### 10.61.1 Detailed Description

Grammar for the FRAT5 curve parser.

Definition at line 89 of file FRAT5ParserHelper.hpp.

#### 10.61.2 Constructor & Destructor Documentation

##### 10.61.2.1 AIRINV::FRAT5ParserHelper::FRAT5Parser::FRAT5Parser (stdair::BomRoot &, [FRAT5Struct](#) &)

Definition at line 115 of file FRAT5ParserHelper.cpp.

### 10.61.3 Member Data Documentation

#### 10.61.3.1 stdair::BomRoot& [AIRINV::FRAT5ParserHelper::FRAT5Parser::\\_bomRoot](#)

Definition at line 107 of file FRAT5ParserHelper.hpp.

Referenced by `AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::definition()`.

#### 10.61.3.2 [FRAT5Struct](#)& [AIRINV::FRAT5ParserHelper::FRAT5Parser::\\_frat5](#)

Definition at line 108 of file FRAT5ParserHelper.hpp.

Referenced by `AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::definition()`.

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

- `airinv/command/FRAT5ParserHelper.hpp`
- `airinv/command/FRAT5ParserHelper.cpp`

## 10.62 AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT > Struct Template Reference

```
#include <airinv/command/FRAT5ParserHelper.hpp>
```

### Public Member Functions

- [definition](#) ([FRAT5Parser](#) const &self)
- `boost::spirit::classic::rule< ScannerT > const & start () const`

### Public Attributes

- `boost::spirit::classic::rule< ScannerT > curve\_list`
- `boost::spirit::classic::rule< ScannerT > not\_to\_be\_parsed`
- `boost::spirit::classic::rule< ScannerT > curve`
- `boost::spirit::classic::rule< ScannerT > key`
- `boost::spirit::classic::rule< ScannerT > map`
- `boost::spirit::classic::rule< ScannerT > value\_pair`
- `boost::spirit::classic::rule< ScannerT > curve\_end`

### 10.62.1 Detailed Description

```
template<typename ScannerT> struct AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >
```

Definition at line 95 of file FRAT5ParserHelper.hpp.

### 10.62.2 Constructor & Destructor Documentation

#### 10.62.2.1 `template<typename ScannerT> AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::definition` ([FRAT5Parser](#) const &self)

Definition at line 124 of file FRAT5ParserHelper.cpp.



References AIRINV::FRAT5ParserHelper::FRAT5Parser::\_bomRoot, AIRINV::FRAT5ParserHelper::FRAT5Parser::\_frat5, AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::curve, AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::curve\_end, AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::curve\_list, AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::key, AIRINV::FRAT5ParserHelper::key\_p(), AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::map, AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::not\_to\_be\_parsed, and AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::value\_pair.

### 10.62.3 Member Function Documentation

#### 10.62.3.1 `template<typename ScannerT> bsc::rule< ScannerT > const & AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::start () const`

Entry point of the parser.

Definition at line 164 of file FRAT5ParserHelper.cpp.

References AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::curve\_list.

### 10.62.4 Member Data Documentation

#### 10.62.4.1 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::curve_list`

Definition at line 99 of file FRAT5ParserHelper.hpp.

Referenced by AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::definition(), and AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::start().

#### 10.62.4.2 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::not_to_be_parsed`

Definition at line 99 of file FRAT5ParserHelper.hpp.

Referenced by AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::definition().

#### 10.62.4.3 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::curve`

Definition at line 99 of file FRAT5ParserHelper.hpp.

Referenced by AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::definition().

#### 10.62.4.4 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::key`

Definition at line 99 of file FRAT5ParserHelper.hpp.

Referenced by AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::definition().

#### 10.62.4.5 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::map`

Definition at line 99 of file FRAT5ParserHelper.hpp.

Referenced by AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::definition().

**10.62.4.6** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::value_pair`

Definition at line 99 of file FRAT5ParserHelper.hpp.

Referenced by AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::definition().

**10.62.4.7** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::curve_end`

Definition at line 99 of file FRAT5ParserHelper.hpp.

Referenced by AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >::definition().

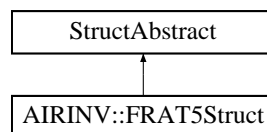
The documentation for this struct was generated from the following files:

- airinv/command/FRAT5ParserHelper.hpp
- airinv/command/FRAT5ParserHelper.cpp

## 10.63 AIRINV::FRAT5Struct Struct Reference

```
#include <airinv/bom/FRAT5Struct.hpp>
```

Inheritance diagram for AIRINV::FRAT5Struct::



### Public Member Functions

- `const std::string describe () const`
- `FRAT5Struct ()`
- `~FRAT5Struct ()`

### Public Attributes

- `std::string _key`
- `stdair::FRAT5Curve_T _curve`
- `stdair::DTD_T _dtd`

### 10.63.1 Detailed Description

Utility Structure for the parsing of FRAT5 structures.

Definition at line 15 of file FRAT5Struct.hpp.

### 10.63.2 Constructor & Destructor Documentation

#### 10.63.2.1 AIRINV::FRAT5Struct::FRAT5Struct ()

Default constructor.

Definition at line 15 of file FRAT5Struct.cpp.

#### 10.63.2.2 AIRINV::FRAT5Struct::~~FRAT5Struct ()

Destructor

Definition at line 19 of file FRAT5Struct.cpp.

### 10.63.3 Member Function Documentation

#### 10.63.3.1 const std::string AIRINV::FRAT5Struct::describe () const

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

Definition at line 23 of file FRAT5Struct.cpp.

References `_curve`, and `_key`.

Referenced by `AIRINV::FRAT5ParserHelper::doEndCurve::operator()()`.

### 10.63.4 Member Data Documentation

#### 10.63.4.1 std::string AIRINV::FRAT5Struct::\_key

Curve key.

Definition at line 37 of file FRAT5Struct.hpp.

Referenced by `describe()`, `AIRINV::FRAT5ParserHelper::doEndCurve::operator()()`, and `AIRINV::FRAT5ParserHelper::storeCurveKey::operator()()`.

#### 10.63.4.2 stdair::FRAT5Curve\_T AIRINV::FRAT5Struct::\_curve

Curve.

Definition at line 40 of file FRAT5Struct.hpp.

Referenced by `describe()`, `AIRINV::FRAT5ParserHelper::doEndCurve::operator()()`, and `AIRINV::FRAT5ParserHelper::storeFRAT5Value::operator()()`.

#### 10.63.4.3 stdair::DTD\_T AIRINV::FRAT5Struct::\_dtd

DTD.

Definition at line 45 of file FRAT5Struct.hpp.

Referenced by `AIRINV::FRAT5ParserHelper::storeFRAT5Value::operator()()`, and `AIRINV::FRAT5ParserHelper::storeDTD::operator()()`.

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

- [airinv/bom/FRAT5Struct.hpp](#)
- [airinv/bom/FRAT5Struct.cpp](#)

## 10.64 AIRINV::header Struct Reference

```
#include <airinv/server/header.hpp>
```

### Public Attributes

- std::string [name](#)
- std::string [value](#)

### 10.64.1 Detailed Description

Header structure.

Definition at line 13 of file header.hpp.

### 10.64.2 Member Data Documentation

#### 10.64.2.1 std::string [AIRINV::header::name](#)

Definition at line 14 of file header.hpp.

#### 10.64.2.2 std::string [AIRINV::header::value](#)

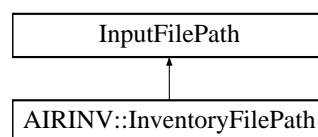
Definition at line 15 of file header.hpp.

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

- airinv/server/[header.hpp](#)

## 10.65 InputFilePath Class Reference

Inheritance diagram for InputFilePath::



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

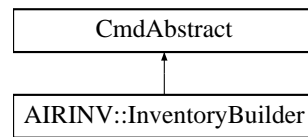
- airinv/[AIRINV\\_Types.hpp](#)

## 10.66 AIRINV::InventoryBuilder Class Reference

Class handling the generation / instantiation of the Inventory BOM.

```
#include <airinv/command/InventoryBuilder.hpp>
```

Inheritance diagram for AIRINV::InventoryBuilder::



## Friends

- class [AIRINV\\_Service](#)
- struct [InventoryParserHelper::doEndFlightDate](#)

### 10.66.1 Detailed Description

Class handling the generation / instantiation of the Inventory BOM.

Definition at line 45 of file InventoryBuilder.hpp.

### 10.66.2 Friends And Related Function Documentation

#### 10.66.2.1 friend class [AIRINV\\_Service](#) [friend]

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

Definition at line 51 of file InventoryBuilder.hpp.

#### 10.66.2.2 friend struct [InventoryParserHelper::doEndFlightDate](#) [friend]

Definition at line 52 of file InventoryBuilder.hpp.

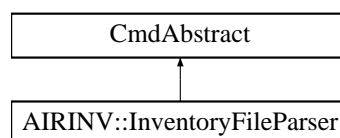
The documentation for this class was generated from the following files:

- airinv/command/[InventoryBuilder.hpp](#)
- airinv/command/[InventoryBuilder.cpp](#)

## 10.67 AIRINV::InventoryFileParser Class Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryFileParser::



## Public Member Functions

- [InventoryFileParser](#) (stdair::BomRoot &, const stdair::Filename\_T &iInventoryInputFilename)
- bool [buildInventory](#) ()

### 10.67.1 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 516 of file InventoryParserHelper.hpp.

### 10.67.2 Constructor & Destructor Documentation

#### 10.67.2.1 AIRINV::InventoryFileParser::InventoryFileParser (stdair::BomRoot &, const stdair::Filename\_T & iInventoryInputFilename)

Constructor.

### 10.67.3 Member Function Documentation

#### 10.67.3.1 bool AIRINV::InventoryFileParser::buildInventory ()

Parse the inventory input file.

Definition at line 1156 of file InventoryParserHelper.cpp.

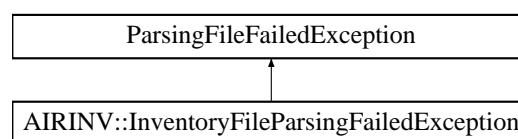
The documentation for this class was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.68 AIRINV::InventoryFileParsingFailedException Class Reference

```
#include <airinv/AIRINV_Types.hpp>
```

Inheritance diagram for AIRINV::InventoryFileParsingFailedException::



### Public Member Functions

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

### 10.68.1 Detailed Description

The inventory input file can not be parsed.

Definition at line 28 of file AIRINV\_Types.hpp.

## 10.68.2 Constructor & Destructor Documentation

### 10.68.2.1 AIRINV::InventoryFileParsingFailedException::InventoryFileParsingFailedException (const std::string & *iWhat*) [inline]

Constructor.

Definition at line 34 of file AIRINV\_Types.hpp.

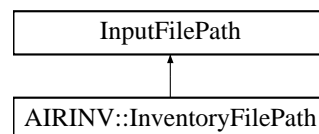
The documentation for this class was generated from the following file:

- [airinv/AIRINV\\_Types.hpp](#)

## 10.69 AIRINV::InventoryFilePath Class Reference

```
#include <airinv/AIRINV_Types.hpp>
```

Inheritance diagram for AIRINV::InventoryFilePath::



## Public Member Functions

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

## 10.69.1 Detailed Description

Inventory input file.

Definition at line 198 of file AIRINV\_Types.hpp.

## 10.69.2 Constructor & Destructor Documentation

### 10.69.2.1 AIRINV::InventoryFilePath::InventoryFilePath (const stdair::Filename\_T & *iFilename*) [inline, explicit]

Constructor.

Definition at line 203 of file AIRINV\_Types.hpp.

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

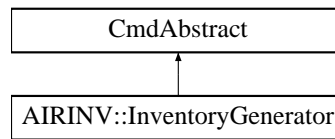
- [airinv/AIRINV\\_Types.hpp](#)

## 10.70 AIRINV::InventoryGenerator Class Reference

Class handling the generation / instantiation of the Inventory BOM.

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

Inheritance diagram for AIRINV::InventoryGenerator::



### Friends

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

#### 10.70.1 Detailed Description

Class handling the generation / instantiation of the Inventory BOM.

Definition at line 42 of file InventoryGenerator.hpp.

#### 10.70.2 Friends And Related Function Documentation

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

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

Definition at line 48 of file InventoryGenerator.hpp.

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

Definition at line 49 of file InventoryGenerator.hpp.

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

Definition at line 50 of file InventoryGenerator.hpp.

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

Definition at line 51 of file InventoryGenerator.hpp.

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

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

### 10.71 AIRINV::InventoryHelper Class Reference

```
#include <airinv/bom/InventoryHelper.hpp>
```



**Static Public Member Functions**

- static void [fillFromRouting](#) (const stdair::Inventory &)
- static void [calculateAvailability](#) (const stdair::Inventory &, const std::string &, stdair::TravelSolutionStruct &)
- static void [getYieldAndBidPrice](#) (const stdair::Inventory &, const std::string &, stdair::TravelSolutionStruct &)
- static bool [sell](#) (stdair::Inventory &, const std::string & iSegmentDateKey, const stdair::ClassCode\_T &, const stdair::PartySize\_T &)
- static bool [sell](#) (const stdair::BookingClassID\_T &, const stdair::PartySize\_T &)
- static bool [cancel](#) (stdair::Inventory &, const std::string & iSegmentDateKey, const stdair::ClassCode\_T &, const stdair::PartySize\_T &)
- static bool [cancel](#) (const stdair::BookingClassID\_T &, const stdair::PartySize\_T &)
- static void [takeSnapshots](#) (const stdair::Inventory &, const stdair::DateTime\_T &)

**10.71.1 Detailed Description**

Class representing the actual business functions for an airline inventory.

Definition at line 23 of file InventoryHelper.hpp.

**10.71.2 Member Function Documentation****10.71.2.1 void AIRINV::InventoryHelper::fillFromRouting (const stdair::Inventory &)**  
[static]

Fill the attributes derived from the routing legs (e.g., board and off dates).

Definition at line 28 of file InventoryHelper.cpp.

References AIRINV::FlightDateHelper::fillFromRouting().

Referenced by AIRINV::BomRootHelper::fillFromRouting().

**10.71.2.2 void AIRINV::InventoryHelper::calculateAvailability (const stdair::Inventory &, const std::string &, stdair::TravelSolutionStruct &)** [static]

Compute the availability for the given travel solution.

Definition at line 44 of file InventoryHelper.cpp.

**10.71.2.3 void AIRINV::InventoryHelper::getYieldAndBidPrice (const stdair::Inventory &, const std::string &, stdair::TravelSolutionStruct &)** [static]

Get yield and bid price information for the given travel solution.

Definition at line 107 of file InventoryHelper.cpp.

**10.71.2.4 bool AIRINV::InventoryHelper::sell (stdair::Inventory &, const std::string & iSegmentDateKey, const stdair::ClassCode\_T &, const stdair::PartySize\_T &)** [static]

Make a sale.

Definition at line 248 of file InventoryHelper.cpp.

Referenced by sell().

**10.71.2.5** `bool AIRINV::InventoryHelper::sell (const stdair::BookingClassID_T &, const stdair::PartySize_T &) [static]`

Make a sale.

Definition at line 282 of file InventoryHelper.cpp.

References `sell()`.

**10.71.2.6** `bool AIRINV::InventoryHelper::cancel (stdair::Inventory &, const std::string & i-SegmentDateKey, const stdair::ClassCode_T &, const stdair::PartySize_T &) [static]`

Make a cancellation.

Definition at line 328 of file InventoryHelper.cpp.

Referenced by `cancel()`.

**10.71.2.7** `bool AIRINV::InventoryHelper::cancel (const stdair::BookingClassID_T &, const stdair::PartySize_T &) [static]`

Make a cancellation.

Definition at line 362 of file InventoryHelper.cpp.

References `cancel()`.

**10.71.2.8** `void AIRINV::InventoryHelper::takeSnapshots (const stdair::Inventory &, const stdair::DateTime_T &) [static]`

Take inventory snapshots.

Definition at line 407 of file InventoryHelper.cpp.

References `AIRINV::SegmentSnapshotTableHelper::takeSnapshots()`.

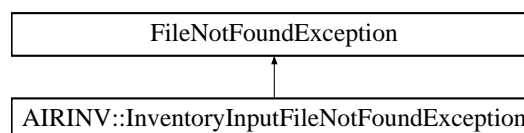
The documentation for this class was generated from the following files:

- [airinv/bom/InventoryHelper.hpp](#)
- [airinv/bom/InventoryHelper.cpp](#)

## 10.72 AIRINV::InventoryInputFileNotFoundException Class Reference

```
#include <airinv/AIRINV_Types.hpp>
```

Inheritance diagram for `AIRINV::InventoryInputFileNotFoundException`:



### Public Member Functions

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

### 10.72.1 Detailed Description

The inventory input file can not be found or opened.

Definition at line 106 of file AIRINV\_Types.hpp.

### 10.72.2 Constructor & Destructor Documentation

#### 10.72.2.1 AIRINV::InventoryInputFileNotFoundException::InventoryInputFileNotFoundException (const std::string & *iWhat*) [inline]

Constructor.

Definition at line 111 of file AIRINV\_Types.hpp.

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

- [airinv/AIRINV\\_Types.hpp](#)

## 10.73 AIRINV::InventoryManager Class Reference

```
#include <airinv/command/InventoryManager.hpp>
```

### Static Public Member Functions

- static void [createDirectAccesses](#) (const stdair::BomRoot &)
- static void [createDirectAccesses](#) (const stdair::BomRoot &, stdair::Inventory &)
- static void [createDirectAccesses](#) (const stdair::BomRoot &, stdair::Inventory &, stdair::FlightDate &)
- static void [createDirectAccesses](#) (stdair::SegmentDate &)
- static void [createPartnerAccesses](#) (const stdair::BomRoot &, stdair::Inventory &)
- static void [createPartnerAccesses](#) (stdair::FlightDate &)
- static void [createPartnerAccesses](#) (const stdair::BomRoot &, stdair::Inventory &, stdair::FlightDate &)
- static void [buildSimilarSegmentCabinSets](#) (const stdair::BomRoot &)
- static void [buildSimilarSegmentCabinSets](#) (stdair::Inventory &)
- static void [buildSegmentSnapshotTable](#) (stdair::Inventory &, const stdair::TableID\_T &, const [DepartureDateSegmentCabinMap\\_T](#) &)
- static void [setDefaultBidPriceVector](#) (stdair::BomRoot &)
- static void [setDefaultBidPriceVector](#) (stdair::Inventory &)
- static void [initialiseYieldBasedNestingStructures](#) (const stdair::BomRoot &)
- static void [initialiseListsOfUsablePolicies](#) (const stdair::BomRoot &)

### Friends

- class [AIRINV\\_Master\\_Service](#)
- class [AIRINV\\_Service](#)

### 10.73.1 Detailed Description

Command wrapping the travel request process.

Definition at line 36 of file InventoryManager.hpp.

### 10.73.2 Member Function Documentation

#### 10.73.2.1 void AIRINV::InventoryManager::createDirectAccesses (const stdair::BomRoot &) [static]

Create the direct accesses within the inventories such as links between leg-date and segment-date, ect.

Definition at line 746 of file InventoryManager.cpp.

References createPartnerAccesses(), and AIRINV::BomRootHelper::fillFromRouting().

Referenced by AIRINV::AIRINV\_Service::buildComplementaryLinks(), and createDirectAccesses().

#### 10.73.2.2 void AIRINV::InventoryManager::createDirectAccesses (const stdair::BomRoot &, stdair::Inventory &) [static]

Definition at line 776 of file InventoryManager.cpp.

References createDirectAccesses().

#### 10.73.2.3 void AIRINV::InventoryManager::createDirectAccesses (const stdair::BomRoot &, stdair::Inventory &, stdair::FlightDate &) [static]

Definition at line 811 of file InventoryManager.cpp.

References createDirectAccesses().

#### 10.73.2.4 void AIRINV::InventoryManager::createDirectAccesses (stdair::SegmentDate &) [static]

Definition at line 860 of file InventoryManager.cpp.

#### 10.73.2.5 void AIRINV::InventoryManager::createPartnerAccesses (const stdair::BomRoot &, stdair::Inventory &) [static]

Create the direct accesses within the inventories such as the link between a marketing segment date and its operating one.

Definition at line 926 of file InventoryManager.cpp.

Referenced by createDirectAccesses().

#### 10.73.2.6 static void AIRINV::InventoryManager::createPartnerAccesses (stdair::FlightDate &) [static]

#### 10.73.2.7 void AIRINV::InventoryManager::createPartnerAccesses (const stdair::BomRoot &, stdair::Inventory &, stdair::FlightDate &) [static]

Definition at line 945 of file InventoryManager.cpp.

#### 10.73.2.8 void AIRINV::InventoryManager::buildSimilarSegmentCabinSets (const stdair::BomRoot &) [static]

Build the similar segment-cabin sets and the corresponding snapshot tables and other data.

Definition at line 1027 of file InventoryManager.cpp.

Referenced by AIRINV::AIRINV\_Service::buildComplementaryLinks().

**10.73.2.9** void AIRINV::InventoryManager::buildSimilarSegmentCabinSets (stdair::Inventory &) [static]

Definition at line 1043 of file InventoryManager.cpp.

References buildSegmentSnapshotTable().

**10.73.2.10** void AIRINV::InventoryManager::buildSegmentSnapshotTable (stdair::Inventory &, const stdair::TableID\_T &, const [DepartureDateSegmentCabinMap\\_T](#) &) [static]

Definition at line 1118 of file InventoryManager.cpp.

Referenced by buildSimilarSegmentCabinSets().

**10.73.2.11** void AIRINV::InventoryManager::setDefaultBidPriceVector (stdair::BomRoot &) [static]

Bid price vectors initialisation

Definition at line 601 of file InventoryManager.cpp.

Referenced by AIRINV::AIRINV\_Service::buildComplementaryLinks().

**10.73.2.12** void AIRINV::InventoryManager::setDefaultBidPriceVector (stdair::Inventory &) [static]

Definition at line 633 of file InventoryManager.cpp.

**10.73.2.13** void AIRINV::InventoryManager::initialiseYieldBasedNestingStructures (const stdair::BomRoot &) [static]

Yield-based nesting structure initialisation

Definition at line 1277 of file InventoryManager.cpp.

References AIRINV::SegmentCabinHelper::initYieldBasedNestingStructure().

Referenced by AIRINV::AIRINV\_Service::buildComplementaryLinks().

**10.73.2.14** void AIRINV::InventoryManager::initialiseListsOfUsablePolicies (const stdair::BomRoot &) [static]

Lists of usable policies initialisation.

Definition at line 1327 of file InventoryManager.cpp.

References AIRINV::SegmentCabinHelper::initListOfUsablePolicies().

Referenced by AIRINV::AIRINV\_Service::buildComplementaryLinks().

### 10.73.3 Friends And Related Function Documentation

**10.73.3.1** friend class [AIRINV\\_Master\\_Service](#) [friend]

Definition at line 37 of file InventoryManager.hpp.

**10.73.3.2** friend class [AIRINV\\_Service](#) [friend]

Definition at line 38 of file InventoryManager.hpp.

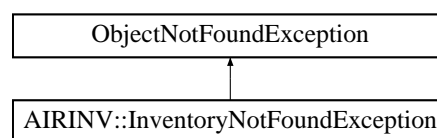
The documentation for this class was generated from the following files:

- [airinv/command/InventoryManager.hpp](#)
- [airinv/command/InventoryManager.cpp](#)

## 10.74 AIRINV::InventoryNotFoundException Class Reference

```
#include <airinv/AIRINV_Types.hpp>
```

Inheritance diagram for AIRINV::InventoryNotFoundException::



### Public Member Functions

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

#### 10.74.1 Detailed Description

Inventory not found.

Definition at line 172 of file AIRINV\_Types.hpp.

#### 10.74.2 Constructor & Destructor Documentation

**10.74.2.1 AIRINV::InventoryNotFoundException::InventoryNotFoundException** (const std::string &*iWhat*) [inline]

Constructor.

Definition at line 177 of file AIRINV\_Types.hpp.

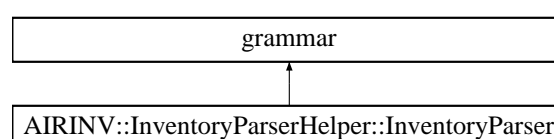
The documentation for this class was generated from the following file:

- [airinv/AIRINV\\_Types.hpp](#)

## 10.75 AIRINV::InventoryParserHelper::InventoryParser Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::InventoryParser::



## Public Member Functions

- [InventoryParser](#) (stdair::BomRoot &, [FlightDateStruct](#) &, unsigned int &)

## Public Attributes

- stdair::BomRoot & [\\_bomRoot](#)
- [FlightDateStruct](#) & [\\_flightDate](#)
- unsigned int & [\\_nbOfFlights](#)

## Classes

- struct [definition](#)

### 10.75.1 Detailed Description

Grammar for the inventory parser.

Definition at line 470 of file InventoryParserHelper.hpp.

### 10.75.2 Constructor & Destructor Documentation

#### 10.75.2.1 AIRINV::InventoryParserHelper::InventoryParser::InventoryParser (stdair::BomRoot &, [FlightDateStruct](#) &, unsigned int &)

Definition at line 894 of file InventoryParserHelper.cpp.

### 10.75.3 Member Data Documentation

#### 10.75.3.1 stdair::BomRoot& [AIRINV::InventoryParserHelper::InventoryParser::\\_bomRoot](#)

Definition at line 498 of file InventoryParserHelper.hpp.

Referenced by [AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition\(\)](#).

#### 10.75.3.2 [FlightDateStruct](#)& [AIRINV::InventoryParserHelper::InventoryParser::\\_flightDate](#)

Definition at line 499 of file InventoryParserHelper.hpp.

Referenced by [AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition\(\)](#).

#### 10.75.3.3 unsigned int& [AIRINV::InventoryParserHelper::InventoryParser::\\_nbOfFlights](#)

Definition at line 500 of file InventoryParserHelper.hpp.

Referenced by [AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition\(\)](#).

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

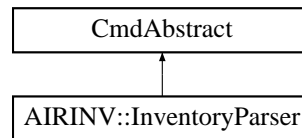
- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.76 AIRINV::InventoryParser Class Reference

Class wrapping the parser entry point.

```
#include <airinv/command/InventoryParser.hpp>
```

Inheritance diagram for AIRINV::InventoryParser::



### Static Public Member Functions

- static void [buildInventory](#) (const [InventoryFilePath](#) &, stdair::BomRoot &)

#### 10.76.1 Detailed Description

Class wrapping the parser entry point.

Definition at line 23 of file InventoryParser.hpp.

#### 10.76.2 Member Function Documentation

**10.76.2.1** void AIRINV::InventoryParser::buildInventory (const [InventoryFilePath](#) &, stdair::BomRoot &) [static]

Parses the CSV file describing an airline inventory, and generates the corresponding data model in memory. It can then be used, for instance, in a simulator.

#### Parameters:

- const* [InventoryFilePath](#)& The file-name of the CSV-formatted inventory input file.
- stdair::BomRoot*& Root of the BOM tree.

Definition at line 20 of file InventoryParser.cpp.

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

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

- airinv/command/[InventoryParser.hpp](#)
- airinv/command/[InventoryParser.cpp](#)

## 10.77 AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT > Struct Template Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```



## Public Member Functions

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

## Public Attributes

- `boost::spirit::classic::rule< ScannerT > flight\_date\_list`
- `boost::spirit::classic::rule< ScannerT > not\_to\_be\_parsed`
- `boost::spirit::classic::rule< ScannerT > flight\_date`
- `boost::spirit::classic::rule< ScannerT > flight\_date\_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 > flight\_type\_code`
- `boost::spirit::classic::rule< ScannerT > flight\_visibility\_code`
- `boost::spirit::classic::rule< ScannerT > date`
- `boost::spirit::classic::rule< ScannerT > leg\_list`
- `boost::spirit::classic::rule< ScannerT > leg`
- `boost::spirit::classic::rule< ScannerT > operating\_leg\_details`
- `boost::spirit::classic::rule< ScannerT > leg\_key`
- `boost::spirit::classic::rule< ScannerT > leg\_details`
- `boost::spirit::classic::rule< ScannerT > leg\_cabin\_list`
- `boost::spirit::classic::rule< ScannerT > leg\_cabin\_details`
- `boost::spirit::classic::rule< ScannerT > bucket\_list`
- `boost::spirit::classic::rule< ScannerT > bucket\_details`
- `boost::spirit::classic::rule< ScannerT > time`
- `boost::spirit::classic::rule< ScannerT > segment\_list`
- `boost::spirit::classic::rule< ScannerT > segment`
- `boost::spirit::classic::rule< ScannerT > segment\_key`
- `boost::spirit::classic::rule< ScannerT > full\_segment\_cabin\_details`
- `boost::spirit::classic::rule< ScannerT > segment\_cabin\_list`
- `boost::spirit::classic::rule< ScannerT > segment\_cabin\_key`
- `boost::spirit::classic::rule< ScannerT > segment\_cabin\_details`
- `boost::spirit::classic::rule< ScannerT > class\_list`
- `boost::spirit::classic::rule< ScannerT > class\_key`
- `boost::spirit::classic::rule< ScannerT > parent\_subclass\_code`
- `boost::spirit::classic::rule< ScannerT > class\_protection`
- `boost::spirit::classic::rule< ScannerT > class\_nego`
- `boost::spirit::classic::rule< ScannerT > class\_details`
- `boost::spirit::classic::rule< ScannerT > family\_cabin\_list`
- `boost::spirit::classic::rule< ScannerT > family\_cabin\_details`

### 10.77.1 Detailed Description

`template<typename ScannerT> struct AIRINV::InventoryParserHelper::Inventory-Parser::definition< ScannerT >`

Definition at line 476 of file `InventoryParserHelper.hpp`.

## 10.77.2 Constructor & Destructor Documentation

### 10.77.2.1 `template<typename ScannerT> AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition (InventoryParser const & self)`

Definition at line 904 of file InventoryParserHelper.cpp.

References AIRINV::InventoryParserHelper::InventoryParser::\_bomRoot, AIRINV::InventoryParserHelper::InventoryParser::\_flightDate, AIRINV::InventoryParserHelper::InventoryParser::\_nbOfFlights, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::airline\_code, AIRINV::InventoryParserHelper::InventoryParser::airline\_code\_p(), AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::bucket\_details, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::bucket\_list, AIRINV::InventoryParserHelper::InventoryParser::cabin\_code\_p(), AIRINV::InventoryParserHelper::class\_code\_list\_p(), AIRINV::InventoryParserHelper::class\_code\_p(), AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::class\_details, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::class\_key, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::class\_list, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::class\_nego, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::class\_protection, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::date, AIRINV::InventoryParserHelper::day\_p(), AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::family\_cabin\_details, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::family\_cabin\_list, AIRINV::InventoryParserHelper::family\_code\_p, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::flight\_date, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::flight\_date\_end, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::flight\_date\_list, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::flight\_key, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::flight\_number, AIRINV::InventoryParserHelper::flight\_number\_p(), AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::flight\_type\_code, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::flight\_visibility\_code, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::full\_segment\_cabin\_details, AIRINV::InventoryParserHelper::hours\_p(), AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::leg, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::leg\_cabin\_details, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::leg\_cabin\_list, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::leg\_details, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::leg\_key, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::leg\_list, AIRINV::InventoryParserHelper::minutes\_p(), AIRINV::InventoryParserHelper::month\_p(), AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::not\_to\_be\_parsed, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::operating\_leg\_details, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::parent\_subclass\_code, AIRINV::InventoryParserHelper::seconds\_p(), AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::segment, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::segment\_cabin\_details, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::segment\_cabin\_key, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::segment\_cabin\_list, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::segment\_key, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::segment\_list, AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::time, AIRINV::InventoryParserHelper::uint1\_2\_p, AIRINV::InventoryParserHelper::uint1\_3\_p, and AIRINV::InventoryParserHelper::year\_p().

### 10.77.3 Member Function Documentation

#### 10.77.3.1 `template<typename ScannerT> bsc::rule< ScannerT > const & AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::start () const`

Entry point of the parser.

Definition at line 1118 of file InventoryParserHelper.cpp.

References AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::flight\_date\_list.

### 10.77.4 Member Data Documentation

#### 10.77.4.1 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::flight_date_list`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition(), and AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::start().

#### 10.77.4.2 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::not_to_be_parsed`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

#### 10.77.4.3 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::flight_date`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

#### 10.77.4.4 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::flight_date_end`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

#### 10.77.4.5 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::flight_key`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

#### 10.77.4.6 `template<typename ScannerT> boost::spirit::classic::rule<ScannerT> AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::airline_code`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.7** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::flight_number`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.8** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::flight_type_code`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.9** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::flight_visibility_  
code`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.10** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::date`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.11** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::leg_list`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.12** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::leg`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.13** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::operating_leg_  
details`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.14** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::leg_key`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.15** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::leg_details`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.16** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::leg_cabin_list`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.17** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::leg_cabin_details`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.18** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::bucket_list`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.19** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::bucket_details`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.20** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::time`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.21** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::segment_list`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.22** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::segment`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.23** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::segment_key`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.24** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::full_segment_  
cabin_details`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.25** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::segment_cabin_  
list`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.26** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::segment_cabin_  
key`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.27** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::segment_cabin_  
details`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.28** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::class_list`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition().

**10.77.4.29** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::class_key`

Definition at line 480 of file InventoryParserHelper.hpp.

Referenced by `AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition()`.

**10.77.4.30** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::parent_subclass_  
code`

Definition at line 480 of file `InventoryParserHelper.hpp`.

Referenced by `AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition()`.

**10.77.4.31** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::class_protection`

Definition at line 480 of file `InventoryParserHelper.hpp`.

Referenced by `AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition()`.

**10.77.4.32** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::class_nego`

Definition at line 480 of file `InventoryParserHelper.hpp`.

Referenced by `AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition()`.

**10.77.4.33** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::class_details`

Definition at line 480 of file `InventoryParserHelper.hpp`.

Referenced by `AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition()`.

**10.77.4.34** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::family_cabin_list`

Definition at line 480 of file `InventoryParserHelper.hpp`.

Referenced by `AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition()`.

**10.77.4.35** `template<typename ScannerT> boost::spirit::classic::rule<ScannerT>  
AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::family_cabin_details`

Definition at line 480 of file `InventoryParserHelper.hpp`.

Referenced by `AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >::definition()`.

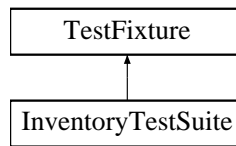
The documentation for this struct was generated from the following files:

- `airinv/command/InventoryParserHelper.hpp`
- `airinv/command/InventoryParserHelper.cpp`

## 10.78 InventoryTestSuite Class Reference

```
#include <test/airinv/InventoryTestSuite.hpp>
```

Inheritance diagram for InventoryTestSuite::



### Public Member Functions

- void [simpleInventory](#) ()
- [InventoryTestSuite](#) ()

### Protected Attributes

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

#### 10.78.1 Detailed Description

Utility class for CPPUNIT-based testing.

Definition at line 7 of file `InventoryTestSuite.hpp`.

#### 10.78.2 Constructor & Destructor Documentation

##### 10.78.2.1 `InventoryTestSuite::InventoryTestSuite ()`

Constructor.

#### 10.78.3 Member Function Documentation

##### 10.78.3.1 `void InventoryTestSuite::simpleInventory ()`

Test a simple inventory functionality.

#### 10.78.4 Member Data Documentation

##### 10.78.4.1 `std::stringstream` [InventoryTestSuite::\\_describeKey](#) [protected]

Definition at line 28 of file `InventoryTestSuite.hpp`.

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

- `test/airinv/InventoryTestSuite.hpp`

## 10.79 AIRINV::LegCabinHelper Class Reference

```
#include <airinv/bom/LegCabinHelper.hpp>
```



### 10.79.1 Detailed Description

Class representing the actual business functions for an airline leg-cabin.

Definition at line 16 of file LegCabinHelper.hpp.

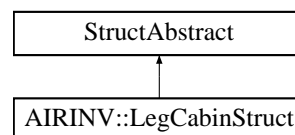
The documentation for this class was generated from the following file:

- [airinv/bom/LegCabinHelper.hpp](#)

## 10.80 AIRINV::LegCabinStruct Struct Reference

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

Inheritance diagram for AIRINV::LegCabinStruct::



### Public Member Functions

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

### Public Attributes

- stdair::CabinCode\_T [\\_cabinCode](#)
- stdair::CabinCapacity\_T [\\_saleableCapacity](#)
- stdair::CapacityAdjustment\_T [\\_adjustment](#)
- stdair::CapacityAdjustment\_T [\\_dcsRegrade](#)
- stdair::AuthorizationLevel\_T [\\_au](#)
- stdair::Availability\_T [\\_avPool](#)
- stdair::UPR\_T [\\_upr](#)
- stdair::NbOfBookings\_T [\\_nbOfBookings](#)
- stdair::Availability\_T [\\_nav](#)
- stdair::Availability\_T [\\_gav](#)
- stdair::OverbookingRate\_T [\\_acp](#)
- stdair::NbOfBookings\_T [\\_etb](#)
- stdair::NbOfBookings\_T [\\_staffNbOfBookings](#)
- stdair::NbOfBookings\_T [\\_wlNbOfBookings](#)
- stdair::NbOfBookings\_T [\\_groupNbOfBookings](#)
- [BucketStructList\\_T \\_bucketList](#)

### 10.80.1 Detailed Description

Utility Structure for the parsing of LegCabin details.

Definition at line 24 of file LegCabinStruct.hpp.

## 10.80.2 Member Function Documentation

### 10.80.2.1 void AIRINV::LegCabinStruct::fill (stdair::LegCabin &) const

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

Definition at line 38 of file LegCabinStruct.cpp.

References [\\_saleableCapacity](#).

### 10.80.2.2 const std::string AIRINV::LegCabinStruct::describe () const

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

Definition at line 15 of file LegCabinStruct.cpp.

References [\\_acp](#), [\\_adjustment](#), [\\_au](#), [\\_avPool](#), [\\_bucketList](#), [\\_cabinCode](#), [\\_dcsRegrade](#), [\\_etb](#), [\\_gav](#), [\\_groupNbOfBookings](#), [\\_nav](#), [\\_nbOfBookings](#), [\\_saleableCapacity](#), [\\_staffNbOfBookings](#), [\\_upr](#), and [\\_wINbOfBookings](#).

## 10.80.3 Member Data Documentation

### 10.80.3.1 stdair::CabinCode\_T AIRINV::LegCabinStruct::\_cabinCode

Definition at line 26 of file LegCabinStruct.hpp.

Referenced by [describe\(\)](#), [AIRINV::ScheduleParserHelper::storeLegCabinCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator\(\)](#), [AIRINV::InventoryParserHelper::storeLegCabinCode::operator\(\)](#), and [AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator\(\)](#).

### 10.80.3.2 stdair::CabinCapacity\_T AIRINV::LegCabinStruct::\_saleableCapacity

Definition at line 27 of file LegCabinStruct.hpp.

Referenced by [describe\(\)](#), [fill\(\)](#), [AIRINV::ScheduleParserHelper::storeCapacity::operator\(\)](#), and [AIRINV::InventoryParserHelper::storeSaleableCapacity::operator\(\)](#).

### 10.80.3.3 stdair::CapacityAdjustment\_T AIRINV::LegCabinStruct::\_adjustment

Definition at line 28 of file LegCabinStruct.hpp.

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

### 10.80.3.4 stdair::CapacityAdjustment\_T AIRINV::LegCabinStruct::\_dcsRegrade

Definition at line 29 of file LegCabinStruct.hpp.

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

### 10.80.3.5 stdair::AuthorizationLevel\_T AIRINV::LegCabinStruct::\_au

Definition at line 30 of file LegCabinStruct.hpp.

Referenced by [describe\(\)](#), and [AIRINV::InventoryParserHelper::storeAU::operator\(\)](#).

**10.80.3.6 stdair::Availability\_T AIRINV::LegCabinStruct::\_avPool**

Definition at line 31 of file LegCabinStruct.hpp.

Referenced by describe().

**10.80.3.7 stdair::UPR\_T AIRINV::LegCabinStruct::\_upr**

Definition at line 32 of file LegCabinStruct.hpp.

Referenced by describe(), and AIRINV::InventoryParserHelper::storeUPR::operator().

**10.80.3.8 stdair::NbOfBookings\_T AIRINV::LegCabinStruct::\_nbOfBookings**

Definition at line 33 of file LegCabinStruct.hpp.

Referenced by describe(), and AIRINV::InventoryParserHelper::storeBookingCounter::operator().

**10.80.3.9 stdair::Availability\_T AIRINV::LegCabinStruct::\_nav**

Definition at line 34 of file LegCabinStruct.hpp.

Referenced by describe(), and AIRINV::InventoryParserHelper::storeNAV::operator().

**10.80.3.10 stdair::Availability\_T AIRINV::LegCabinStruct::\_gav**

Definition at line 35 of file LegCabinStruct.hpp.

Referenced by describe(), and AIRINV::InventoryParserHelper::storeGAV::operator().

**10.80.3.11 stdair::OverbookingRate\_T AIRINV::LegCabinStruct::\_acp**

Definition at line 36 of file LegCabinStruct.hpp.

Referenced by describe(), and AIRINV::InventoryParserHelper::storeACP::operator().

**10.80.3.12 stdair::NbOfBookings\_T AIRINV::LegCabinStruct::\_etb**

Definition at line 37 of file LegCabinStruct.hpp.

Referenced by describe(), and AIRINV::InventoryParserHelper::storeETB::operator().

**10.80.3.13 stdair::NbOfBookings\_T AIRINV::LegCabinStruct::\_staffNbOfBookings**

Definition at line 38 of file LegCabinStruct.hpp.

Referenced by describe().

**10.80.3.14 stdair::NbOfBookings\_T AIRINV::LegCabinStruct::\_wlNbOfBookings**

Definition at line 39 of file LegCabinStruct.hpp.

Referenced by describe().

**10.80.3.15 stdair::NbOfBookings\_T AIRINV::LegCabinStruct::\_groupNbOfBookings**

Definition at line 40 of file LegCabinStruct.hpp.

Referenced by describe().

### 10.80.3.16 BucketStructList\_T AIRINV::LegCabinStruct::\_bucketList

Definition at line 41 of file LegCabinStruct.hpp.

Referenced by describe(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator>(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator>(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator>(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator>(), and AIRINV::InventoryParserHelper::storeAirlineCode::operator().

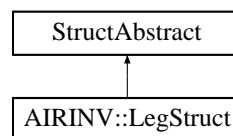
The documentation for this struct was generated from the following files:

- [airinv/bom/LegCabinStruct.hpp](#)
- [airinv/bom/LegCabinStruct.cpp](#)

## 10.81 AIRINV::LegStruct Struct Reference

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

Inheritance diagram for AIRINV::LegStruct::



### Public Member Functions

- void [fill](#) (const stdair::Date\_T &iRefDate, stdair::LegDate &) const
- void [fill](#) (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::Date\_T [\\_boardingDate](#)
- stdair::Duration\_T [\\_boardingTime](#)
- stdair::AirportCode\_T [\\_offPoint](#)
- stdair::DateOffset\_T [\\_offDateOffset](#)
- stdair::Date\_T [\\_offDate](#)
- stdair::Duration\_T [\\_offTime](#)
- stdair::Duration\_T [\\_elapsed](#)
- [LegCabinStructList\\_T](#) [\\_cabinList](#)

### 10.81.1 Detailed Description

Utility Structure for the parsing of Leg structures.

Definition at line 24 of file LegStruct.hpp.

### 10.81.2 Constructor & Destructor Documentation

#### 10.81.2.1 AIRINV::LegStruct::LegStruct ()

Default Constructor.

Definition at line 16 of file LegStruct.cpp.

### 10.81.3 Member Function Documentation

#### 10.81.3.1 void AIRINV::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 41 of file LegStruct.cpp.

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

#### 10.81.3.2 void AIRINV::LegStruct::fill (stdair::LegDate &) const

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

Definition at line 62 of file LegStruct.cpp.

References `_airlineCode`, `_boardingTime`, `_elapsed`, `_flightNumber`, `_offDate`, `_offPoint`, and `_offTime`.

#### 10.81.3.3 const std::string AIRINV::LegStruct::describe () const

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

Definition at line 21 of file LegStruct.cpp.

References `_boardingDate`, `_boardingPoint`, `_boardingTime`, `_cabinList`, `_elapsed`, `_offDate`, `_offPoint`, and `_offTime`.

### 10.81.4 Member Data Documentation

#### 10.81.4.1 stdair::AirlineCode\_T AIRINV::LegStruct::\_airlineCode

Definition at line 26 of file LegStruct.hpp.

Referenced by `fill()`, `AIRINV::ScheduleParserHelper::storeOperatingAirlineCode::operator()()`, `AIRINV::ScheduleParserHelper::storeDateRangeEnd::operator()()`, `AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator()()`, and `AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator()()`.

**10.81.4.2 stdair::FlightNumber\_T AIRINV::LegStruct::\_flightNumber**

Definition at line 27 of file LegStruct.hpp.

Referenced by fill(), AIRINV::ScheduleParserHelper::storeOperatingFlightNumber::operator>(), AIRINV::ScheduleParserHelper::storeDateRangeEnd::operator>(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), and AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator().

**10.81.4.3 stdair::AirportCode\_T AIRINV::LegStruct::\_boardingPoint**

Definition at line 28 of file LegStruct.hpp.

Referenced by describe(), AIRINV::ScheduleParserHelper::storeLegBoardingPoint::operator(), and AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator().

**10.81.4.4 stdair::DateOffset\_T AIRINV::LegStruct::\_boardingDateOffset**

Definition at line 29 of file LegStruct.hpp.

Referenced by fill(), and AIRINV::ScheduleParserHelper::storeOffTime::operator().

**10.81.4.5 stdair::Date\_T AIRINV::LegStruct::\_boardingDate**

Definition at line 30 of file LegStruct.hpp.

Referenced by describe(), and AIRINV::InventoryParserHelper::storeBoardingDate::operator().

**10.81.4.6 stdair::Duration\_T AIRINV::LegStruct::\_boardingTime**

Definition at line 31 of file LegStruct.hpp.

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

**10.81.4.7 stdair::AirportCode\_T AIRINV::LegStruct::\_offPoint**

Definition at line 32 of file LegStruct.hpp.

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

**10.81.4.8 stdair::DateOffset\_T AIRINV::LegStruct::\_offDateOffset**

Definition at line 33 of file LegStruct.hpp.

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

**10.81.4.9 stdair::Date\_T AIRINV::LegStruct::\_offDate**

Definition at line 34 of file LegStruct.hpp.

Referenced by describe(), fill(), and AIRINV::InventoryParserHelper::storeOffDate::operator().

**10.81.4.10 stdair::Duration\_T AIRINV::LegStruct::\_offTime**

Definition at line 35 of file LegStruct.hpp.

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

#### 10.81.4.11 `std::duration<T>` AIRINV::LegStruct::\_elapsed

Definition at line 36 of file `LegStruct.hpp`.

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

#### 10.81.4.12 `LegCabinStructList_T` AIRINV::LegStruct::\_cabinList

Definition at line 37 of file `LegStruct.hpp`.

Referenced by `describe()`, `AIRINV::ScheduleParserHelper::doEndFlight::operator()`, `AIRINV::ScheduleParserHelper::storeCapacity::operator()`, `AIRINV::ScheduleParserHelper::storeLegBoardingPoint::operator()`, `AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator()`, `AIRINV::InventoryParserHelper::storeLegCabinCode::operator()`, `AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator()`, and `AIRINV::InventoryParserHelper::storeAirlineCode::operator()`.

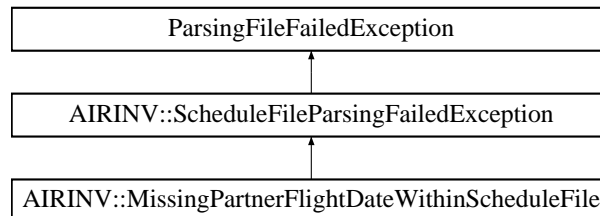
The documentation for this struct was generated from the following files:

- `airinv/bom/LegStruct.hpp`
- `airinv/bom/LegStruct.cpp`

## 10.82 AIRINV::MissingPartnerFlightDateWithinScheduleFile Class Reference

```
#include <airinv/AIRINV_Types.hpp>
```

Inheritance diagram for `AIRINV::MissingPartnerFlightDateWithinScheduleFile`:



### Public Member Functions

- `MissingPartnerFlightDateWithinScheduleFile` (`const std::string &iWhat`)

#### 10.82.1 Detailed Description

Missing partner flight date within the schedule file.

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

#### 10.82.2 Constructor & Destructor Documentation

##### 10.82.2.1 `AIRINV::MissingPartnerFlightDateWithinScheduleFile::MissingPartnerFlightDateWithinScheduleFile` (`const std::string &iWhat`) `[inline]`

Constructor.

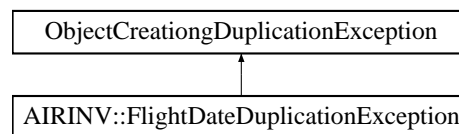
Definition at line 60 of file AIRINV\_Types.hpp.

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

- [airinv/AIRINV\\_Types.hpp](#)

## 10.83 ObjectCreationgDuplicationException Class Reference

Inheritance diagram for ObjectCreationgDuplicationException::

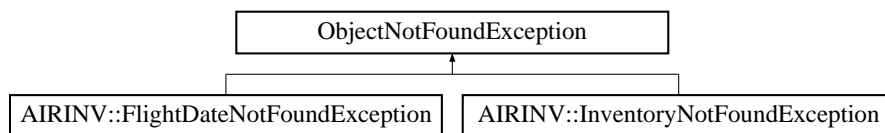


The documentation for this class was generated from the following file:

- [airinv/AIRINV\\_Types.hpp](#)

## 10.84 ObjectNotFoundException Class Reference

Inheritance diagram for ObjectNotFoundException::

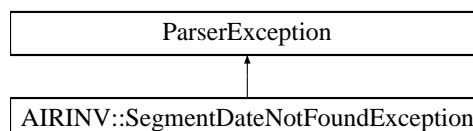


The documentation for this class was generated from the following file:

- [airinv/AIRINV\\_Types.hpp](#)

## 10.85 ParserException Class Reference

Inheritance diagram for ParserException::



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

- [airinv/AIRINV\\_Types.hpp](#)



```
#include <airinv/command/InventoryParserHelper.hpp>
```

[illegible]

- ParserSemanticAction (FlightDateStruct &)

- FlightDateStruct & \_flightDate

Generic Semantic Action (Actor / Functor) for the Inventory Parser.  
Definition at line 29 of file InventoryParserHelper.hpp.

## 10.86.2 Constructor & Destructor Documentation

### 10.86.2.1 AIRINV::InventoryParserHelper::ParserSemanticAction::ParserSemanticAction (FlightDateStruct &)

Actor Constructor.

Definition at line 26 of file InventoryParserHelper.cpp.

## 10.86.3 Member Data Documentation

### 10.86.3.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

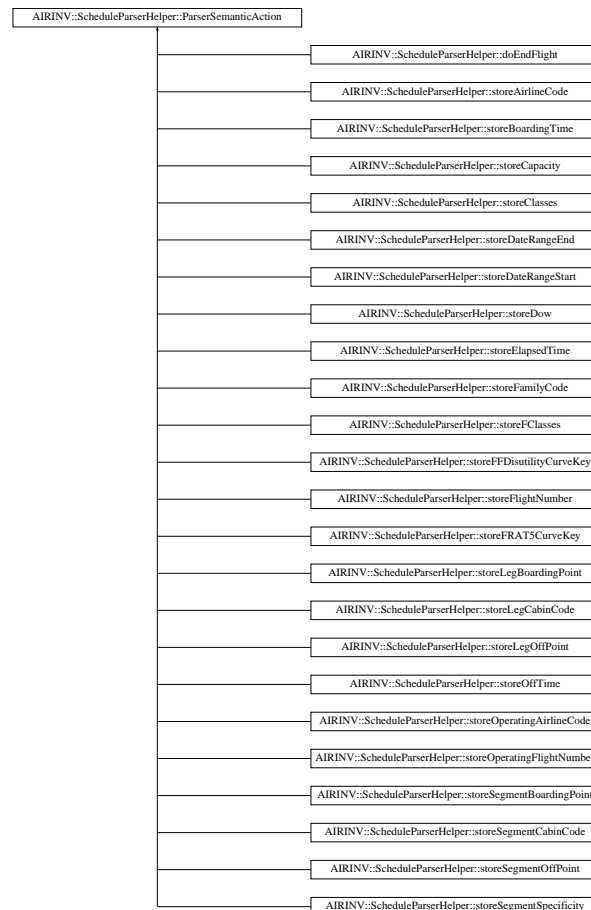
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.87 AIRINV::ScheduleParserHelper::ParserSemanticAction Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::ParserSemanticAction::



### Public Member Functions

- [ParserSemanticAction](#) ([FlightPeriodStruct](#) &)

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.87.1 Detailed Description

Generic Semantic Action (Actor / Functor) for the Schedule Parser.

Definition at line 29 of file `ScheduleParserHelper.hpp`.

### 10.87.2 Constructor & Destructor Documentation

#### 10.87.2.1 AIRINV::ScheduleParserHelper::ParserSemanticAction::ParserSemanticAction (FlightPeriodStruct &)

Actor Constructor.

Definition at line 28 of file ScheduleParserHelper.cpp.

### 10.87.3 Member Data Documentation

#### 10.87.3.1 FlightPeriodStruct& AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flight-Period

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

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

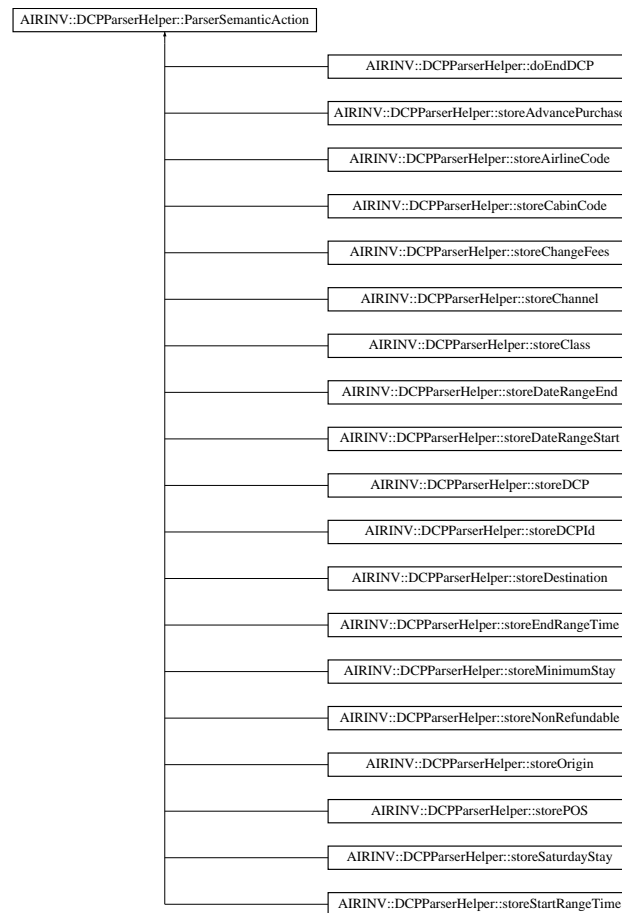
The documentation for this struct was generated from the following files:

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

## 10.88 AIRINV::DCPParserHelper::ParserSemanticAction Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::ParserSemanticAction::



## Public Member Functions

- [ParserSemanticAction](#) (DCPRuleStruct &)

## Public Attributes

- DCPRuleStruct & [\\_DCPRule](#)

### 10.88.1 Detailed Description

Generic Semantic Action (Actor / Functor) for the DCP Parser.

Definition at line 30 of file DCPParserHelper.hpp.

### 10.88.2 Constructor & Destructor Documentation

#### 10.88.2.1 AIRINV::DCPParserHelper::ParserSemanticAction::ParserSemanticAction (DCPRuleStruct &)

Actor Constructor.

Definition at line 25 of file DCPParserHelper.cpp.

### 10.88.3 Member Data Documentation

#### 10.88.3.1 DCPRuleStruct& AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule

Actor Context.

Definition at line 34 of file DCPParserHelper.hpp.

Referenced by AIRINV::DCPParserHelper::doEndDCP::operator(), AIRINV::DCPParserHelper::storeClass::operator(), AIRINV::DCPParserHelper::storeAirlineCode::operator(), AIRINV::DCPParserHelper::storeDCP::operator(), AIRINV::DCPParserHelper::storeMinimumStay::operator(), AIRINV::DCPParserHelper::storeNonRefundable::operator(), AIRINV::DCPParserHelper::storeChangeFees::operator(), AIRINV::DCPParserHelper::storeSaturdayStay::operator(), AIRINV::DCPParserHelper::storeAdvancePurchase::operator(), AIRINV::DCPParserHelper::storeChannel::operator(), AIRINV::DCPParserHelper::storeCabinCode::operator(), AIRINV::DCPParserHelper::storePOS::operator(), AIRINV::DCPParserHelper::storeEndRangeTime::operator(), AIRINV::DCPParserHelper::storeStartRangeTime::operator(), AIRINV::DCPParserHelper::storeDateRangeEnd::operator(), AIRINV::DCPParserHelper::storeDateRangeStart::operator(), AIRINV::DCPParserHelper::storeDestination::operator(), AIRINV::DCPParserHelper::storeOrigin::operator(), and AIRINV::DCPParserHelper::storeDCPID::operator().

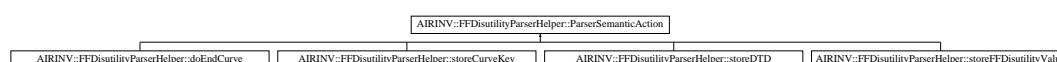
The documentation for this struct was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.89 AIRINV::FFDisutilityParserHelper::ParserSemanticAction Struct Reference

```
#include <airinv/command/FFDisutilityParserHelper.hpp>
```

Inheritance diagram for AIRINV::FFDisutilityParserHelper::ParserSemanticAction::



### Public Member Functions

- [ParserSemanticAction](#) (FFDisutilityStruct &)

### Public Attributes

- [FFDisutilityStruct](#) & [\\_ffDisutility](#)

### 10.89.1 Detailed Description

Generic Semantic Action (Actor / Functor) for the FFDisutility Parser.

Definition at line 29 of file FFDisutilityParserHelper.hpp.

## 10.89.2 Constructor & Destructor Documentation

### 10.89.2.1 AIRINV::FFDisutilityParserHelper::ParserSemanticAction::ParserSemanticAction (FFDisutilityStruct &)

Actor Constructor.

Definition at line 27 of file FFDisutilityParserHelper.cpp.

## 10.89.3 Member Data Documentation

### 10.89.3.1 FFDisutilityStruct& AIRINV::FFDisutilityParserHelper::ParserSemanticAction::\_ffDisutility

Actor Context.

Definition at line 33 of file FFDisutilityParserHelper.hpp.

Referenced by AIRINV::FFDisutilityParserHelper::doEndCurve::operator>(), AIRINV::FFDisutilityParserHelper::storeFFDisutilityValue::operator>(), AIRINV::FFDisutilityParserHelper::storeDTD::operator>(), and AIRINV::FFDisutilityParserHelper::storeCurveKey::operator().

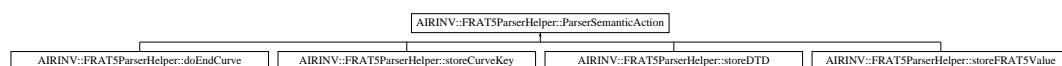
The documentation for this struct was generated from the following files:

- [airinv/command/FFDisutilityParserHelper.hpp](#)
- [airinv/command/FFDisutilityParserHelper.cpp](#)

## 10.90 AIRINV::FRAT5ParserHelper::ParserSemanticAction Struct Reference

```
#include <airinv/command/FRAT5ParserHelper.hpp>
```

Inheritance diagram for AIRINV::FRAT5ParserHelper::ParserSemanticAction::



## Public Member Functions

- [ParserSemanticAction \(FRAT5Struct &\)](#)

## Public Attributes

- [FRAT5Struct & \\_frat5](#)

### 10.90.1 Detailed Description

Generic Semantic Action (Actor / Functor) for the FRAT5 Parser.

Definition at line 29 of file FRAT5ParserHelper.hpp.

## 10.90.2 Constructor & Destructor Documentation

### 10.90.2.1 AIRINV::FRAT5ParserHelper::ParserSemanticAction::ParserSemanticAction (FRAT5Struct &)

Actor Constructor.

Definition at line 27 of file FRAT5ParserHelper.cpp.

## 10.90.3 Member Data Documentation

### 10.90.3.1 FRAT5Struct& AIRINV::FRAT5ParserHelper::ParserSemanticAction::\_frat5

Actor Context.

Definition at line 33 of file FRAT5ParserHelper.hpp.

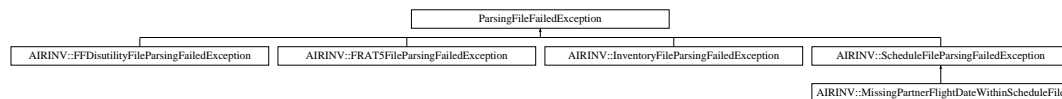
Referenced by AIRINV::FRAT5ParserHelper::doEndCurve::operator(), AIRINV::FRAT5ParserHelper::storeFRAT5Value::operator(), AIRINV::FRAT5ParserHelper::storeDTD::operator(), and AIRINV::FRAT5ParserHelper::storeCurveKey::operator().

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

- [airinv/command/FRAT5ParserHelper.hpp](#)
- [airinv/command/FRAT5ParserHelper.cpp](#)

## 10.91 ParsingFileFailedException Class Reference

Inheritance diagram for ParsingFileFailedException::



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

- [airinv/AIRINV\\_Types.hpp](#)

## 10.92 AIRINV::Reply Struct Reference

```
#include <airinv/server/Reply.hpp>
```

### Public Member Functions

- `std::vector< boost::asio::const_buffer > to_buffers ()`

### Public Attributes

- `FlightRequestStatus::EN_FlightRequestStatus _status`
- `std::string content`



### 10.92.1 Detailed Description

A reply to be sent to a client.

Definition at line 18 of file Reply.hpp.

### 10.92.2 Member Function Documentation

#### 10.92.2.1 `std::vector< boost::asio::const_buffer > AIRINV::Reply::to_buffers ()`

Convert the reply into a vector of buffers. The buffers do not own the underlying memory blocks, therefore the reply object must remain valid and not be changed until the write operation has completed.

Definition at line 15 of file Reply.cpp.

References content.

### 10.92.3 Member Data Documentation

#### 10.92.3.1 `FlightRequestStatus::EN_FlightRequestStatus AIRINV::Reply::_status`

Status.

Definition at line 20 of file Reply.hpp.

Referenced by `AIRINV::RequestHandler::handleRequest()`.

#### 10.92.3.2 `std::string AIRINV::Reply::content`

The content to be sent in the reply.

Definition at line 23 of file Reply.hpp.

Referenced by `AIRINV::RequestHandler::handleRequest()`, and `to_buffers()`.

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

- `airinv/server/Reply.hpp`
- `airinv/server/Reply.cpp`

## 10.93 AIRINV::Request Struct Reference

```
#include <airinv/server/Request.hpp>
```

### Public Member Functions

- `bool parseFlightDate ()`

### Public Attributes

- `std::string _flightDetails`
- `stdair::AirlineCode_T _airlineCode`
- `stdair::FlightNumber_T _flightNumber`
- `stdair::Date_T _departureDate`

### 10.93.1 Detailed Description

A request received from a client.

Definition at line 18 of file Request.hpp.

### 10.93.2 Member Function Documentation

#### 10.93.2.1 bool AIRINV::Request::parseFlightDate ()

Parse the incoming request.

Expected requested is of the form: <airline\_code>,<flight\_number>,<flight\_date>, where date format is YYYY-MM-DD. For instance: BA,341,2010-09-20.

Definition at line 12 of file Request.cpp.

References `_airlineCode`, `_departureDate`, and `_flightNumber`.

Referenced by `AIRINV::RequestHandler::handleRequest()`.

### 10.93.3 Member Data Documentation

#### 10.93.3.1 std::string AIRINV::Request::\_flightDetails

String as it comes from the connected client.

Definition at line 29 of file Request.hpp.

Referenced by `AIRINV::RequestHandler::handleRequest()`.

#### 10.93.3.2 stdair::AirlineCode\_T AIRINV::Request::\_airlineCode

Parsed airline code.

Definition at line 31 of file Request.hpp.

Referenced by `parseFlightDate()`.

#### 10.93.3.3 stdair::FlightNumber\_T AIRINV::Request::\_flightNumber

Parsed flight number.

Definition at line 33 of file Request.hpp.

Referenced by `parseFlightDate()`.

#### 10.93.3.4 stdair::Date\_T AIRINV::Request::\_departureDate

Parsed departure date.

Definition at line 35 of file Request.hpp.

Referenced by `parseFlightDate()`.

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

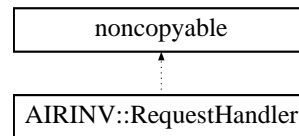
- [airinv/server/Request.hpp](#)
- [airinv/server/Request.cpp](#)

## 10.94 AIRINV::RequestHandler Class Reference

The common handler for all incoming requests.

```
#include <airinv/server/RequestHandler.hpp>
```

Inheritance diagram for AIRINV::RequestHandler::



### Public Member Functions

- [RequestHandler](#) (const stdair::AirlineCode\_T &)
- bool [handleRequest](#) ([Request](#) &, [Reply](#) &) const

#### 10.94.1 Detailed Description

The common handler for all incoming requests.

Definition at line 28 of file RequestHandler.hpp.

#### 10.94.2 Constructor & Destructor Documentation

##### 10.94.2.1 AIRINV::RequestHandler::RequestHandler (const stdair::AirlineCode\_T &)

Constructor.

##### Parameters:

*const* stdair::AirlineCode\_T& Airline code of the inventory owner.

Definition at line 20 of file RequestHandler.cpp.

#### 10.94.3 Member Function Documentation

##### 10.94.3.1 bool AIRINV::RequestHandler::handleRequest ([Request](#) &, [Reply](#) &) const

Handle a request and produce a reply.

Definition at line 26 of file RequestHandler.cpp.

References [AIRINV::Request::\\_flightDetails](#), [AIRINV::Reply::\\_status](#), [AIRINV::Reply::content](#), [AIRINV::FlightRequestStatus::INTERNAL\\_ERROR](#), [AIRINV::FlightRequestStatus::OK](#), and [AIRINV::Request::parseFlightDate\(\)](#).

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

- [airinv/server/RequestHandler.hpp](#)
- [airinv/server/RequestHandler.cpp](#)

## 10.95 AIRINV::RequestParser Class Reference

Parser for incoming requests.

```
#include <airinv/server/RequestParser.hpp>
```

### Public Member Functions

- [RequestParser \(\)](#)  
*Construct ready to parse the request method.*
- void [reset \(\)](#)  
*Reset to initial parser state.*
- template<typename InputIterator> boost::tuple< boost::tribool, InputIterator > [parse \(Request &req, InputIterator begin, InputIterator end\)](#)

### 10.95.1 Detailed Description

Parser for incoming requests.

Definition at line 17 of file RequestParser.hpp.

### 10.95.2 Constructor & Destructor Documentation

#### 10.95.2.1 AIRINV::RequestParser::RequestParser ()

Construct ready to parse the request method.

Definition at line 13 of file RequestParser.cpp.

### 10.95.3 Member Function Documentation

#### 10.95.3.1 void AIRINV::RequestParser::reset ()

Reset to initial parser state.

Definition at line 18 of file RequestParser.cpp.

#### 10.95.3.2 template<typename InputIterator> boost::tuple<boost::tribool, InputIterator> AIRINV::RequestParser::parse ([Request](#) & req, InputIterator begin, InputIterator end) [inline]

Parse some data. The tribool return value is true when a complete request has been parsed, false if the data is invalid, indeterminate when more data is required. The InputIterator return value indicates how much of the input has been consumed.

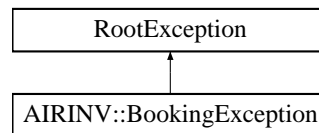
Definition at line 30 of file RequestParser.hpp.

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

- airinv/server/[RequestParser.hpp](#)
- airinv/server/[RequestParser.cpp](#)

## 10.96 RootException Class Reference

Inheritance diagram for RootException::



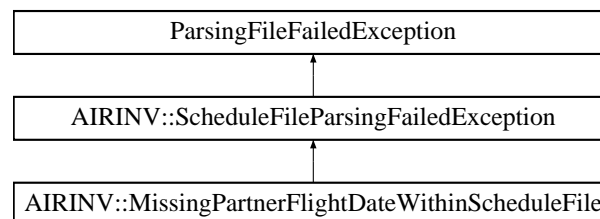
The documentation for this class was generated from the following file:

- [airinv/AIRINV\\_Types.hpp](#)

## 10.97 AIRINV::ScheduleFileParsingFailedException Class Reference

```
#include <airinv/AIRINV_Types.hpp>
```

Inheritance diagram for AIRINV::ScheduleFileParsingFailedException::



### Public Member Functions

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

### 10.97.1 Detailed Description

The schedule input file can not be parsed.

Definition at line 41 of file AIRINV\_Types.hpp.

### 10.97.2 Constructor & Destructor Documentation

#### 10.97.2.1 AIRINV::ScheduleFileParsingFailedException::ScheduleFileParsingFailedException (const std::string &iWhat) [inline]

Constructor.

Definition at line 47 of file AIRINV\_Types.hpp.

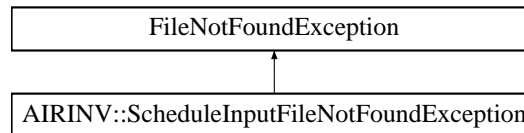
The documentation for this class was generated from the following file:

- [airinv/AIRINV\\_Types.hpp](#)

## 10.98 AIRINV::ScheduleInputFileNotFoundException Class Reference

```
#include <airinv/AIRINV_Types.hpp>
```

Inheritance diagram for AIRINV::ScheduleInputFileNotFoundException::



### Public Member Functions

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

#### 10.98.1 Detailed Description

The schedule input file can not be found or opened.

Definition at line 118 of file AIRINV\_Types.hpp.

#### 10.98.2 Constructor & Destructor Documentation

##### 10.98.2.1 AIRINV::ScheduleInputFileNotFoundException::ScheduleInputFileNotFoundException (const std::string &*iWhat*) [inline]

Constructor.

Definition at line 123 of file AIRINV\_Types.hpp.

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

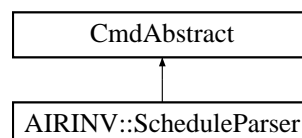
- airinv/[AIRINV\\_Types.hpp](#)

## 10.99 AIRINV::ScheduleParser Class Reference

Class wrapping the parser entry point.

```
#include <airinv/command/ScheduleParser.hpp>
```

Inheritance diagram for AIRINV::ScheduleParser::



### Static Public Member Functions

- static void [generateInventories](#) (const stdair::ScheduleFilePath &iScheduleFilename, stdair::Bom-Root &)

### 10.99.1 Detailed Description

Class wrapping the parser entry point.

Definition at line 22 of file ScheduleParser.hpp.

### 10.99.2 Member Function Documentation

#### 10.99.2.1 void AIRINV::ScheduleParser::generateInventories (const stdair::ScheduleFilePath & i-ScheduleFilename, stdair::BomRoot &) [static]

Parse the CSV file describing the airline schedules for the simulator, and generates the inventories accordingly.

#### Parameters:

- const** stdair::Filename\_T& 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.

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

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

- [airinv/command/ScheduleParser.hpp](#)
- [airinv/command/ScheduleParser.cpp](#)

## 10.100 AIRINV::SegmentCabinHelper Class Reference

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

```
#include <airinv/bom/SegmentCabinHelper.hpp>
```

### Static Public Member Functions

- static void [updateFromReservation](#) (const stdair::FlightDate &, stdair::SegmentCabin &, const stdair::PartySize\_T &)
- static void [buildPseudoBidPriceVector](#) (stdair::SegmentCabin &)
- static void [updateBookingControlsUsingPseudoBidPriceVector](#) (const stdair::SegmentCabin &)
- static void [updateAUs](#) (const stdair::SegmentCabin &)
- static void [updateAvailabilities](#) (const stdair::SegmentCabin &)
- static void [initialiseAU](#) (stdair::SegmentCabin &)
- static void [initYieldBasedNestingStructure](#) (stdair::SegmentCabin &)
- static void [initListOfUsablePolicies](#) (stdair::SegmentCabin &)

### 10.100.1 Detailed Description

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

Definition at line 25 of file SegmentCabinHelper.hpp.

## 10.100.2 Member Function Documentation

### 10.100.2.1 void AIRINV::SegmentCabinHelper::updateFromReservation (const stdair::FlightDate &, stdair::SegmentCabin &, const stdair::PartySize\_T &) [static]

Update the segment-cabin with the reservation.

Definition at line 64 of file SegmentCabinHelper.cpp.

References AIRINV::FlightDateHelper::updateAvailability().

### 10.100.2.2 void AIRINV::SegmentCabinHelper::buildPseudoBidPriceVector (stdair::SegmentCabin &) [static]

Build the pseudo bid price vector from the vectors of the leg-cabins.

Definition at line 77 of file SegmentCabinHelper.cpp.

Referenced by AIRINV::FlightDateHelper::updateBookingControls().

### 10.100.2.3 void AIRINV::SegmentCabinHelper::updateBookingControlsUsingPseudoBidPriceVector (const stdair::SegmentCabin &) [static]

Update the booking controls using the pseudo bid price vector.

Definition at line 128 of file SegmentCabinHelper.cpp.

References updateAUs().

Referenced by AIRINV::FlightDateHelper::updateBookingControls().

### 10.100.2.4 void AIRINV::SegmentCabinHelper::updateAUs (const stdair::SegmentCabin &) [static]

Update the authorisation levels using the booking limits.

Definition at line 186 of file SegmentCabinHelper.cpp.

Referenced by updateBookingControlsUsingPseudoBidPriceVector().

### 10.100.2.5 void AIRINV::SegmentCabinHelper::updateAvailabilities (const stdair::SegmentCabin &) [static]

Update the availability of the booking classes.

Definition at line 240 of file SegmentCabinHelper.cpp.

Referenced by AIRINV::FlightDateHelper::recalculateAvailability(), and AIRINV::SegmentSnapshotTableHelper::takeSnapshots().

### 10.100.2.6 void AIRINV::SegmentCabinHelper::initialiseAU (stdair::SegmentCabin &) [static]

Initialise the AU for the booking classes.

Definition at line 28 of file SegmentCabinHelper.cpp.

Referenced by AIRINV::SegmentDateHelper::fillFromRouting().



**10.100.2.7** void AIRINV::SegmentCabinHelper::initYieldBasedNestingStructure (stdair::SegmentCabin &) [static]

Yield-based nesting structure initialisation.

Definition at line 333 of file SegmentCabinHelper.cpp.

Referenced by AIRINV::InventoryManager::initialiseYieldBasedNestingStructures().

**10.100.2.8** void AIRINV::SegmentCabinHelper::initListOfUsablePolicies (stdair::SegmentCabin &) [static]

List of usable policies initialisation.

Definition at line 385 of file SegmentCabinHelper.cpp.

Referenced by AIRINV::InventoryManager::initialiseListsOfUsablePolicies().

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

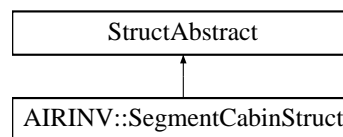
- airinv/bom/[SegmentCabinHelper.hpp](#)
- airinv/bom/[SegmentCabinHelper.cpp](#)

## 10.101 AIRINV::SegmentCabinStruct Struct Reference

Utility Structure for the parsing of SegmentCabin details.

```
#include <airinv/bom/SegmentCabinStruct.hpp>
```

Inheritance diagram for AIRINV::SegmentCabinStruct::



### Public Member Functions

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

### Public Attributes

- stdair::CabinCode\_T [\\_cabinCode](#)
- stdair::NbOfBookings\_T [\\_nbOfBookings](#)
- [FareFamilyStruct](#) [\\_itFareFamily](#)
- [FareFamilyStructList\\_T](#) [\\_fareFamilies](#)

### 10.101.1 Detailed Description

Utility Structure for the parsing of SegmentCabin details.

Definition at line 26 of file SegmentCabinStruct.hpp.

### 10.101.2 Member Function Documentation

#### 10.101.2.1 void AIRINV::SegmentCabinStruct::fill (stdair::SegmentCabin &) const

Fill the SegmentCabin objects with the attributes of the [SegmentCabinStruct](#).

Definition at line 33 of file SegmentCabinStruct.cpp.

#### 10.101.2.2 const std::string AIRINV::SegmentCabinStruct::describe () const

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

Definition at line 15 of file SegmentCabinStruct.cpp.

References `_cabinCode`, `_fareFamilies`, and `_nbOfBookings`.

### 10.101.3 Member Data Documentation

#### 10.101.3.1 stdair::CabinCode\_T AIRINV::SegmentCabinStruct::\_cabinCode

Definition at line 28 of file SegmentCabinStruct.hpp.

Referenced by `AIRINV::FlightPeriodStruct::addFareFamily()`, `AIRINV::FlightDateStruct::addFareFamily()`, `describe()`, `AIRINV::ScheduleParserHelper::storeSegmentCabinCode::operator()`, and `AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator()`.

#### 10.101.3.2 stdair::NbOfBookings\_T AIRINV::SegmentCabinStruct::\_nbOfBookings

Definition at line 29 of file SegmentCabinStruct.hpp.

Referenced by `describe()`, and `AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator()`.

#### 10.101.3.3 FareFamilyStruct AIRINV::SegmentCabinStruct::\_itFareFamily

Definition at line 30 of file SegmentCabinStruct.hpp.

Referenced by `AIRINV::ScheduleParserHelper::storeFCClasses::operator()`, `AIRINV::ScheduleParserHelper::storeFFDisutilityCurveKey::operator()`, `AIRINV::ScheduleParserHelper::storeFRAT5CurveKey::operator()`, `AIRINV::ScheduleParserHelper::storeFamilyCode::operator()`, `AIRINV::ScheduleParserHelper::storeClasses::operator()`, `AIRINV::InventoryParserHelper::storeFCClasses::operator()`, `AIRINV::InventoryParserHelper::storeFamilyCode::operator()`, `AIRINV::InventoryParserHelper::storeClassCode::operator()`, `AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator()`, `AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator()`, and `AIRINV::InventoryParserHelper::storeAirlineCode::operator()`.

#### 10.101.3.4 FareFamilyStructList\_T AIRINV::SegmentCabinStruct::\_fareFamilies

Definition at line 31 of file SegmentCabinStruct.hpp.

Referenced by `describe()`, `AIRINV::InventoryParserHelper::storeFCClasses::operator()`, `AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator()`, `AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator()`, and `AIRINV::InventoryParserHelper::storeAirlineCode::operator()`.

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

- `airinv/bom/SegmentCabinStruct.hpp`

- [airinv/bom/SegmentCabinStruct.cpp](#)

## 10.102 AIRINV::SegmentDateHelper Class Reference

```
#include <airinv/bom/SegmentDateHelper.hpp>
```

### Static Public Member Functions

- static void [fillFromRouting](#) (stdair::SegmentDate &)
- static void [updateElapsedTimeFromRouting](#) (stdair::SegmentDate &)
- static void [updateDistanceFromElapsedTime](#) (stdair::SegmentDate &)

### 10.102.1 Detailed Description

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

Definition at line 16 of file SegmentDateHelper.hpp.

### 10.102.2 Member Function Documentation

#### 10.102.2.1 void AIRINV::SegmentDateHelper::fillFromRouting (stdair::SegmentDate &) [static]

Fill the attributes derived from the routing legs (e.g., board and off dates).

Definition at line 18 of file SegmentDateHelper.cpp.

References AIRINV::SegmentCabinHelper::initialiseAU(), and updateElapsedTimeFromRouting().

Referenced by AIRINV::FlightDateHelper::fillFromRouting().

#### 10.102.2.2 void AIRINV::SegmentDateHelper::updateElapsedTimeFromRouting (stdair::SegmentDate &) [static]

Calculate and set the elapsed time according to the leg routing.

Actually, the elapsed time of the segment is the sum of the elapsed times of the routing legs, plus the stop-over times. The stop-over time is the difference between the board time of a routing leg, and the off time of the previous leg. That is, it is the time spent at the corresponding airport.

Of course, in case of mono-leg segments, there is no stop-over, and the elapsed time of the segment is equal to the elapsed time of the single routing leg.

Definition at line 72 of file SegmentDateHelper.cpp.

References updateDistanceFromElapsedTime().

Referenced by fillFromRouting().

#### 10.102.2.3 void AIRINV::SegmentDateHelper::updateDistanceFromElapsedTime (stdair::SegmentDate &) [static]

Method computing the distance of the segment (in kilometers).

Definition at line 115 of file SegmentDateHelper.cpp.

Referenced by updateElapsedTimeFromRouting().

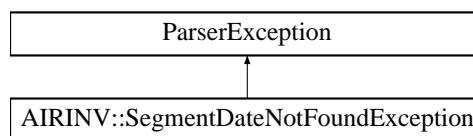
The documentation for this class was generated from the following files:

- [airinv/bom/SegmentDateHelper.hpp](#)
- [airinv/bom/SegmentDateHelper.cpp](#)

### 10.103 AIRINV::SegmentDateNotFoundException Class Reference

```
#include <airinv/AIRINV_Types.hpp>
```

Inheritance diagram for AIRINV::SegmentDateNotFoundException::



#### Public Member Functions

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

#### 10.103.1 Detailed Description

Specific exception when some BOM objects can not be found within the inventory.

Definition at line 94 of file AIRINV\_Types.hpp.

#### 10.103.2 Constructor & Destructor Documentation

**10.103.2.1 AIRINV::SegmentDateNotFoundException::SegmentDateNotFoundException (const std::string &iWhat) [inline]**

Constructor.

Definition at line 99 of file AIRINV\_Types.hpp.

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

- [airinv/AIRINV\\_Types.hpp](#)

### 10.104 AIRINV::SegmentSnapshotTableHelper Class Reference

```
#include <airinv/bom/SegmentSnapshotTableHelper.hpp>
```

#### Static Public Member Functions

- static void [takeSnapshots](#) (stdair::SegmentSnapshotTable &, const stdair::DateTime\_T &)

### 10.104.1 Detailed Description

Class representing the actual business functions for an airline inventory.

Definition at line 22 of file SegmentSnapshotTableHelper.hpp.

### 10.104.2 Member Function Documentation

#### 10.104.2.1 void AIRINV::SegmentSnapshotTableHelper::takeSnapshots (stdair::SegmentSnapshotTable &, const stdair::DateTime\_T &) [static]

Take inventory snapshots.

Definition at line 27 of file SegmentSnapshotTableHelper.cpp.

References AIRINV::SegmentCabinHelper::updateAvailabilities().

Referenced by AIRINV::InventoryHelper::takeSnapshots().

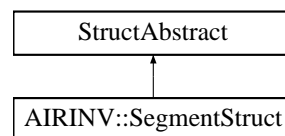
The documentation for this class was generated from the following files:

- [airinv/bom/SegmentSnapshotTableHelper.hpp](#)
- [airinv/bom/SegmentSnapshotTableHelper.cpp](#)

## 10.105 AIRINV::SegmentStruct Struct Reference

```
#include <airinv/bom/SegmentStruct.hpp>
```

Inheritance diagram for AIRINV::SegmentStruct::



### Public Member Functions

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

### Public Attributes

- stdair::AirportCode\_T [\\_boardingPoint](#)
- stdair::AirportCode\_T [\\_offPoint](#)
- stdair::Date\_T [\\_boardingDate](#)
- stdair::Duration\_T [\\_boardingTime](#)
- stdair::Date\_T [\\_offDate](#)
- stdair::Duration\_T [\\_offTime](#)
- stdair::Duration\_T [\\_elapsed](#)
- [SegmentCabinStructList\\_T\\_cabinList](#)

### 10.105.1 Detailed Description

Utility Structure for the parsing of Segment structures.

Definition at line 23 of file SegmentStruct.hpp.

### 10.105.2 Member Function Documentation

#### 10.105.2.1 void AIRINV::SegmentStruct::fill (stdair::SegmentDate &) const

Fill the SegmentDate objects with the attributes of the [SegmentStruct](#).

Definition at line 36 of file SegmentStruct.cpp.

References `_boardingTime`, `_elapsed`, `_offDate`, and `_offTime`.

#### 10.105.2.2 const std::string AIRINV::SegmentStruct::describe () const

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

Definition at line 14 of file SegmentStruct.cpp.

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

### 10.105.3 Member Data Documentation

#### 10.105.3.1 stdair::AirportCode\_T AIRINV::SegmentStruct::\_boardingPoint

Definition at line 25 of file SegmentStruct.hpp.

Referenced by `AIRINV::FlightPeriodStruct::addFareFamily()`, `AIRINV::FlightDateStruct::addFareFamily()`, `AIRINV::FlightPeriodStruct::addSegmentCabin()`, `AIRINV::FlightDateStruct::addSegmentCabin()`, `describe()`, `AIRINV::ScheduleParserHelper::storeSegmentBoardingPoint::operator()`, and `AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator()`.

#### 10.105.3.2 stdair::AirportCode\_T AIRINV::SegmentStruct::\_offPoint

Definition at line 26 of file SegmentStruct.hpp.

Referenced by `AIRINV::FlightPeriodStruct::addFareFamily()`, `AIRINV::FlightDateStruct::addFareFamily()`, `AIRINV::FlightPeriodStruct::addSegmentCabin()`, `AIRINV::FlightDateStruct::addSegmentCabin()`, `describe()`, `AIRINV::ScheduleParserHelper::storeSegmentOffPoint::operator()`, and `AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator()`.

#### 10.105.3.3 stdair::Date\_T AIRINV::SegmentStruct::\_boardingDate

Definition at line 27 of file SegmentStruct.hpp.

#### 10.105.3.4 stdair::Duration\_T AIRINV::SegmentStruct::\_boardingTime

Definition at line 28 of file SegmentStruct.hpp.

Referenced by `describe()`, and `fill()`.

#### 10.105.3.5 stdair::Date\_T AIRINV::SegmentStruct::\_offDate

Definition at line 29 of file SegmentStruct.hpp.

Referenced by `fill()`.

### 10.105.3.6 `stdair::Duration_T` [AIRINV::SegmentStruct::\\_offTime](#)

Definition at line 30 of file `SegmentStruct.hpp`.

Referenced by `describe()`, and `fill()`.

### 10.105.3.7 `stdair::Duration_T` [AIRINV::SegmentStruct::\\_elapsed](#)

Definition at line 31 of file `SegmentStruct.hpp`.

Referenced by `describe()`, and `fill()`.

### 10.105.3.8 [SegmentCabinStructList\\_T](#) [AIRINV::SegmentStruct::\\_cabinList](#)

Definition at line 32 of file `SegmentStruct.hpp`.

Referenced by `describe()`, `AIRINV::InventoryParserHelper::doEndFlightDate::operator()`, `AIRINV::InventoryParserHelper::storeFCClasses::operator()`, `AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator()`, `AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator()`, and `AIRINV::InventoryParserHelper::storeAirlineCode::operator()`.

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

- `airinv/bom/SegmentStruct.hpp`
- `airinv/bom/SegmentStruct.cpp`

## 10.106 AIRINV::ServiceAbstract Class Reference

```
#include <airinv/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()`

### 10.106.1 Detailed Description

Base class for the Service layer.

Definition at line 14 of file `ServiceAbstract.hpp`.

### 10.106.2 Constructor & Destructor Documentation

#### 10.106.2.1 `virtual AIRINV::ServiceAbstract::~~ServiceAbstract()` [`inline`, `virtual`]

Destructor.

Definition at line 18 of file ServiceAbstract.hpp.

#### 10.106.2.2 AIRINV::ServiceAbstract::ServiceAbstract () [inline, protected]

Protected Default Constructor to ensure this class is abstract.

Definition at line 30 of file ServiceAbstract.hpp.

### 10.106.3 Member Function Documentation

#### 10.106.3.1 virtual void AIRINV::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.

#### 10.106.3.2 virtual void AIRINV::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.

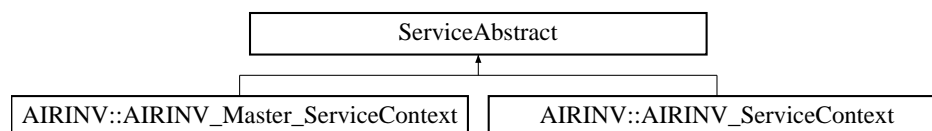
Referenced by operator>>().

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

- [airinv/service/ServiceAbstract.hpp](#)

## 10.107 ServiceAbstract Class Reference

Inheritance diagram for ServiceAbstract::



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

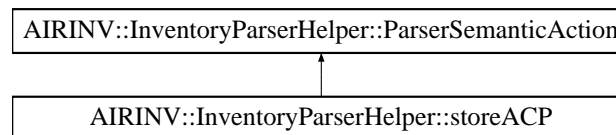
- [airinv/service/AIRINV\\_Master\\_ServiceContext.hpp](#)
- [airinv/service/AIRINV\\_ServiceContext.hpp](#)



## 10.108 AIRINV::InventoryParserHelper::storeACP Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeACP::



### Public Member Functions

- [storeACP](#) ([FlightDateStruct](#) &)
- void [operator\(\)](#) (double *iReal*) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.108.1 Detailed Description

Store the parsed Average Cancellation Percentage (ACP).

Definition at line 205 of file `InventoryParserHelper.hpp`.

#### 10.108.2 Constructor & Destructor Documentation

##### 10.108.2.1 AIRINV::InventoryParserHelper::storeACP::storeACP ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 350 of file `InventoryParserHelper.cpp`.

#### 10.108.3 Member Function Documentation

##### 10.108.3.1 void AIRINV::InventoryParserHelper::storeACP::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 355 of file `InventoryParserHelper.cpp`.

References `AIRINV::LegCabinStruct::_acp`, `AIRINV::InventoryParserHelper::ParserSemanticAction::_flightDate`, and `AIRINV::FlightDateStruct::_itLegCabin`.

#### 10.108.4 Member Data Documentation

##### 10.108.4.1 [FlightDateStruct](#)& [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#) [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

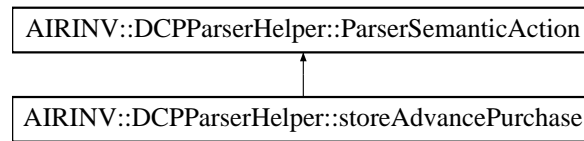
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.109 AIRINV::DCPParserHelper::storeAdvancePurchase Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::storeAdvancePurchase::



## Public Member Functions

- [storeAdvancePurchase](#) (DCPRuleStruct &)
- void [operator\(\)](#) (unsigned int, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

## Public Attributes

- DCPRuleStruct & [\\_DCPRule](#)

### 10.109.1 Detailed Description

Store the parsed advance purchase days.

Definition at line 138 of file DCPParserHelper.hpp.

### 10.109.2 Constructor & Destructor Documentation

#### 10.109.2.1 AIRINV::DCPParserHelper::storeAdvancePurchase::storeAdvancePurchase (DCPRuleStruct &)

Actor Constructor.

Definition at line 208 of file DCPParserHelper.cpp.

### 10.109.3 Member Function Documentation

#### 10.109.3.1 void AIRINV::DCPParserHelper::storeAdvancePurchase::operator() (unsigned int, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

Actor Function (functor).

Definition at line 213 of file DCPParserHelper.cpp.

References AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule.

### 10.109.4 Member Data Documentation

#### 10.109.4.1 DCPRuleStruct& [AIRINV::DCPParserHelper::ParserSemanticAction::\\_DCPRule](#) [inherited]

Actor Context.

Definition at line 34 of file DCPParserHelper.hpp.

Referenced by AIRINV::DCPParserHelper::doEndDCP::operator>(), AIRINV::DCPParserHelper::storeClass::operator>(), AIRINV::DCPParserHelper::storeAirlineCode::operator>(), AIRINV::DCPParserHelper::storeDCP::operator>(), AIRINV::DCPParserHelper::storeMinimumStay::operator>(),

AIRINV::DCPParserHelper::storeNonRefundable::operator(), AIRINV::DCPParserHelper::storeChangeFees::operator(), AIRINV::DCPParserHelper::storeSaturdayStay::operator(), operator(), AIRINV::DCPParserHelper::storeChannel::operator(), AIRINV::DCPParserHelper::storeCabinCode::operator(), AIRINV::DCPParserHelper::storePOS::operator(), AIRINV::DCPParserHelper::storeEndRangeTime::operator(), AIRINV::DCPParserHelper::storeStartRangeTime::operator(), AIRINV::DCPParserHelper::storeDateRangeEnd::operator(), AIRINV::DCPParserHelper::storeDateRangeStart::operator(), AIRINV::DCPParserHelper::storeDestination::operator(), AIRINV::DCPParserHelper::storeOrigin::operator(), and AIRINV::DCPParserHelper::storeDCPId::operator().

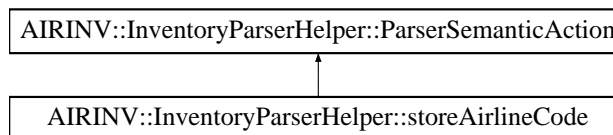
The documentation for this struct was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.110 AIRINV::InventoryParserHelper::storeAirlineCode Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeAirlineCode::



### Public Member Functions

- [storeAirlineCode](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.110.1 Detailed Description

Store the parsed airline code.

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

#### 10.110.2 Constructor & Destructor Documentation

##### 10.110.2.1 AIRINV::InventoryParserHelper::storeAirlineCode::storeAirlineCode ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 44 of file [InventoryParserHelper.cpp](#).

### 10.110.3 Member Function Documentation

#### 10.110.3.1 void AIRINV::InventoryParserHelper::storeAirlineCode::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 49 of file InventoryParserHelper.cpp.

References AIRINV::FlightDateStruct::\_airlineCode, AIRINV::LegCabinStruct::\_bucketList, AIRINV::SegmentStruct::\_cabinList, AIRINV::LegStruct::\_cabinList, AIRINV::BookingClassStruct::\_classCode, AIRINV::FareFamilyStruct::\_classList, AIRINV::SegmentCabinStruct::\_fareFamilies, AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_itBookingClass, AIRINV::FlightDateStruct::\_itBucket, AIRINV::SegmentCabinStruct::\_itFareFamily, AIRINV::FlightDateStruct::\_itLeg, AIRINV::FlightDateStruct::\_itLegCabin, AIRINV::FlightDateStruct::\_itSegment, AIRINV::FlightDateStruct::\_itSegmentCabin, AIRINV::FlightDateStruct::\_legList, AIRINV::FlightDateStruct::\_segmentList, and AIRINV::BucketStruct::\_yieldRangeUpperValue.

### 10.110.4 Member Data Documentation

#### 10.110.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), and AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlight

Number::operator()), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator()).

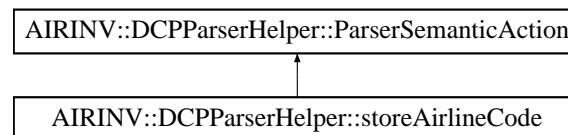
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.111 AIRINV::DCPParserHelper::storeAirlineCode Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::storeAirlineCode::



### Public Member Functions

- [storeAirlineCode](#) (DCPRuleStruct &)
- void [operator\(\)](#) (std::vector< char >, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

### Public Attributes

- DCPRuleStruct & [\\_DCPRule](#)

#### 10.111.1 Detailed Description

Store the parsed airline code.

Definition at line 198 of file DCPParserHelper.hpp.

#### 10.111.2 Constructor & Destructor Documentation

##### 10.111.2.1 AIRINV::DCPParserHelper::storeAirlineCode::storeAirlineCode (DCPRuleStruct &)

Actor Constructor.

Definition at line 329 of file DCPParserHelper.cpp.

### 10.111.3 Member Function Documentation

#### 10.111.3.1 void AIRINV::DCPParserHelper::storeAirlineCode::operator() (std::vector< char >, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

Actor Function (functor).

Definition at line 334 of file DCPParserHelper.cpp.

References AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule.

### 10.111.4 Member Data Documentation

#### 10.111.4.1 DCPRuleStruct& [AIRINV::DCPParserHelper::ParserSemanticAction::\\_DCPRule](#) [inherited]

Actor Context.

Definition at line 34 of file DCPParserHelper.hpp.

Referenced by AIRINV::DCPParserHelper::doEndDCP::operator(), AIRINV::DCPParserHelper::storeClass::operator(), operator(), AIRINV::DCPParserHelper::storeDCP::operator(), AIRINV::DCPParserHelper::storeMinimumStay::operator(), AIRINV::DCPParserHelper::storeNonRefundable::operator(), AIRINV::DCPParserHelper::storeChangeFees::operator(), AIRINV::DCPParserHelper::storeSaturdayStay::operator(), AIRINV::DCPParserHelper::storeAdvancePurchase::operator(), AIRINV::DCPParserHelper::storeChannel::operator(), AIRINV::DCPParserHelper::storeCabinCode::operator(), AIRINV::DCPParserHelper::storePOS::operator(), AIRINV::DCPParserHelper::storeEndRangeTime::operator(), AIRINV::DCPParserHelper::storeStartRangeTime::operator(), AIRINV::DCPParserHelper::storeDateRangeEnd::operator(), AIRINV::DCPParserHelper::storeDateRangeStart::operator(), AIRINV::DCPParserHelper::storeDestination::operator(), AIRINV::DCPParserHelper::storeOrigin::operator(), and AIRINV::DCPParserHelper::storeDCPID::operator().

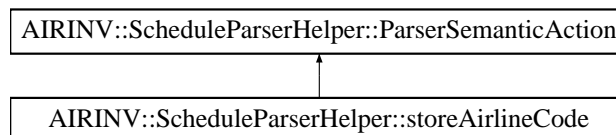
The documentation for this struct was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.112 AIRINV::ScheduleParserHelper::storeAirlineCode Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeAirlineCode::



### Public Member Functions

- [storeAirlineCode](#) ([FlightPeriodStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

**Public Attributes**

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

**10.112.1 Detailed Description**

Store the parsed airline code.

Definition at line 37 of file ScheduleParserHelper.hpp.

**10.112.2 Constructor & Destructor Documentation****10.112.2.1 AIRINV::ScheduleParserHelper::storeAirlineCode::storeAirlineCode ([FlightPeriodStruct](#) &)**

Actor Constructor.

Definition at line 34 of file ScheduleParserHelper.cpp.

**10.112.3 Member Function Documentation****10.112.3.1 void AIRINV::ScheduleParserHelper::storeAirlineCode::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const**

Actor Function (functor).

Definition at line 39 of file ScheduleParserHelper.cpp.

References [AIRINV::FlightPeriodStruct::\\_airlineCode](#), [AIRINV::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#), and [AIRINV::FlightPeriodStruct::\\_legList](#).

**10.112.4 Member Data Documentation****10.112.4.1 [FlightPeriodStruct](#)& AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod** [[inherited](#)]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

Referenced by [AIRINV::ScheduleParserHelper::doEndFlight::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeFCClasses::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeFFDisutilityCurveKey::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeFRAT5CurveKey::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeFamilyCode::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeClasses::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeSegmentCabinCode::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeSegmentOffPoint::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeSegmentBoardingPoint::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeSegmentSpecificity::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeCapacity::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeLegCabinCode::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeElapsedTime::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeOffTime::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeBoardingTime::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeOperatingFlightNumber::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeOperatingAirlineCode::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeLegOffPoint::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeLegBoardingPoint::operator\(\)](#), [AIRINV::ScheduleParserHelper::storeDow::operator\(\)](#), and [AIRINV::ScheduleParserHelper::storeDateRangeEnd::operator\(\)](#).



AIRINV::ScheduleParserHelper::storeDateRangeStart::operator(),  
 Helper::storeFlightNumber::operator(), and operator().

AIRINV::ScheduleParser-

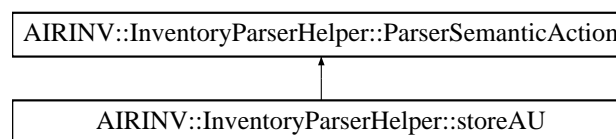
The documentation for this struct was generated from the following files:

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

## 10.113 AIRINV::InventoryParserHelper::storeAU Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeAU::



### Public Member Functions

- [storeAU](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) (double iReal) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.113.1 Detailed Description

Store the parsed Authorisation Level (AU).

Definition at line 165 of file InventoryParserHelper.hpp.

#### 10.113.2 Constructor & Destructor Documentation

##### 10.113.2.1 AIRINV::InventoryParserHelper::storeAU::storeAU ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 295 of file InventoryParserHelper.cpp.

#### 10.113.3 Member Function Documentation

##### 10.113.3.1 void AIRINV::InventoryParserHelper::storeAU::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 300 of file InventoryParserHelper.cpp.

References [AIRINV::LegCabinStruct::\\_au](#), [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), and [AIRINV::FlightDateStruct::\\_itLegCabin](#).

### 10.113.4 Member Data Documentation

#### 10.113.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

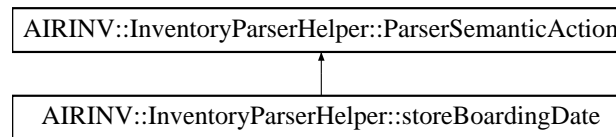
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.114 AIRINV::InventoryParserHelper::storeBoardingDate Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeBoardingDate::



### Public Member Functions

- [storeBoardingDate](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.114.1 Detailed Description

Store the boarding date.

Definition at line 117 of file InventoryParserHelper.hpp.

#### 10.114.2 Constructor & Destructor Documentation

##### 10.114.2.1 AIRINV::InventoryParserHelper::storeBoardingDate::storeBoardingDate ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 204 of file InventoryParserHelper.cpp.

#### 10.114.3 Member Function Documentation

##### 10.114.3.1 void AIRINV::InventoryParserHelper::storeBoardingDate::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 209 of file InventoryParserHelper.cpp.

References [AIRINV::LegStruct::\\_boardingDate](#), [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), [AIRINV::FlightDateStruct::\\_itLeg](#), and [AIRINV::FlightDateStruct::getDate\(\)](#).

#### 10.114.4 Member Data Documentation

##### 10.114.4.1 [FlightDateStruct](#)& [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#) [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

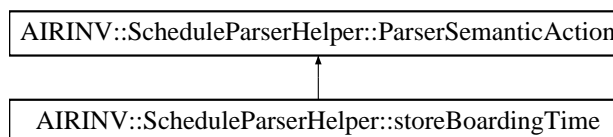
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.115 AIRINV::ScheduleParserHelper::storeBoardingTime Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeBoardingTime::



## Public Member Functions

- [storeBoardingTime](#) ([FlightPeriodStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

## Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

### 10.115.1 Detailed Description

Store the boarding time.

Definition at line 109 of file `ScheduleParserHelper.hpp`.

### 10.115.2 Constructor & Destructor Documentation

#### 10.115.2.1 AIRINV::ScheduleParserHelper::storeBoardingTime::storeBoardingTime ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 190 of file `ScheduleParserHelper.cpp`.

### 10.115.3 Member Function Documentation

#### 10.115.3.1 void AIRINV::ScheduleParserHelper::storeBoardingTime::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 195 of file `ScheduleParserHelper.cpp`.

References `AIRINV::LegStruct::_boardingTime`, `AIRINV::FlightPeriodStruct::_dateOffset`, `AIRINV::ScheduleParserHelper::ParserSemanticAction::_flightPeriod`, `AIRINV::FlightPeriodStruct::_itLeg`, `AIRINV::FlightPeriodStruct::_itSeconds`, and `AIRINV::FlightPeriodStruct::getTime()`.

### 10.115.4 Member Data Documentation

#### 10.115.4.1 [FlightPeriodStruct](#)& AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [inherited]

Actor Context.

Definition at line 33 of file `ScheduleParserHelper.hpp`.

Referenced by `AIRINV::ScheduleParserHelper::doEndFlight::operator()`, `AIRINV::ScheduleParserHelper::storeFClasses::operator()`, `AIRINV::ScheduleParserHelper::storeFFDisutilityCurveKey::operator()`, `AIRINV::ScheduleParserHelper::storeFRAT5CurveKey::operator()`, `AIRINV::ScheduleParserHelper::storeFamilyCode::operator()`, `AIRINV::ScheduleParserHelper::storeClasses::operator()`, `AIRINV::ScheduleParserHelper::storeSegmentCabinCode::operator()`, `AIRINV::ScheduleParserHelper::storeSegmentOffPoint::operator()`, `AIRINV::ScheduleParserHelper::storeSegmentBoardingPoint::operator()`, `AIRINV::ScheduleParserHelper::storeSegmentSpecificity::operator()`, and `AIRINV::ScheduleParserHelper::storeCapacity::operator()`.

AIRINV::ScheduleParserHelper::storeLegCabinCode::operator>(), AIRINV::ScheduleParserHelper::storeElapsedTime::operator>(), AIRINV::ScheduleParserHelper::storeOffTime::operator>(), operator(), AIRINV::ScheduleParserHelper::storeOperatingFlightNumber::operator>(), AIRINV::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRINV::ScheduleParserHelper::storeLegOffPoint::operator(), AIRINV::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRINV::ScheduleParserHelper::storeDow::operator(), AIRINV::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRINV::ScheduleParserHelper::storeDateRangeStart::operator(), AIRINV::ScheduleParserHelper::storeFlightNumber::operator(), and AIRINV::ScheduleParserHelper::storeAirlineCode::operator()).

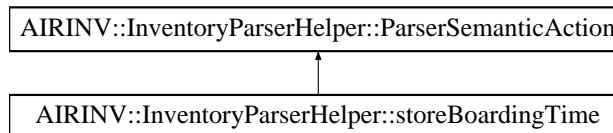
The documentation for this struct was generated from the following files:

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

## 10.116 AIRINV::InventoryParserHelper::storeBoardingTime Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeBoardingTime::



### Public Member Functions

- [storeBoardingTime](#) ([FlightDateStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.116.1 Detailed Description

Store the boarding time.

Definition at line 125 of file [InventoryParserHelper.hpp](#).

#### 10.116.2 Constructor & Destructor Documentation

##### 10.116.2.1 AIRINV::InventoryParserHelper::storeBoardingTime::storeBoardingTime ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 215 of file [InventoryParserHelper.cpp](#).

### 10.116.3 Member Function Documentation

#### 10.116.3.1 void AIRINV::InventoryParserHelper::storeBoardingTime::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 220 of file InventoryParserHelper.cpp.

References AIRINV::LegStruct::\_boardingTime, AIRINV::FlightDateStruct::\_dateOffSet, AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_itLeg, AIRINV::FlightDateStruct::\_itSeconds, and AIRINV::FlightDateStruct::getTime().

### 10.116.4 Member Data Documentation

#### 10.116.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirline

Code::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

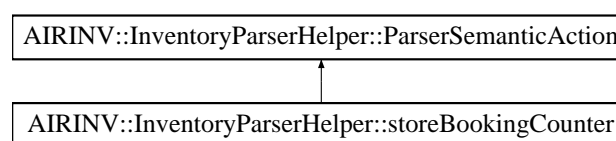
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.117 AIRINV::InventoryParserHelper::storeBookingCounter Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeBookingCounter:



### Public Member Functions

- [storeBookingCounter](#) ([FlightDateStruct](#) &)
- void [operator\(\)](#) (double iReal) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.117.1 Detailed Description

Store the parsed booking counter.

Definition at line 181 of file InventoryParserHelper.hpp.

#### 10.117.2 Constructor & Destructor Documentation

##### 10.117.2.1 AIRINV::InventoryParserHelper::storeBookingCounter::storeBookingCounter ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 317 of file InventoryParserHelper.cpp.

#### 10.117.3 Member Function Documentation

##### 10.117.3.1 void AIRINV::InventoryParserHelper::storeBookingCounter::operator() (double *i-Real*) const

Actor Function (functor).

Definition at line 322 of file InventoryParserHelper.cpp.

References [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), [AIRINV::FlightDateStruct::\\_itLegCabin](#), and [AIRINV::LegCabinStruct::\\_nbOfBookings](#).



#### 10.117.4 Member Data Documentation

##### 10.117.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flight-Date [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvaibility::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

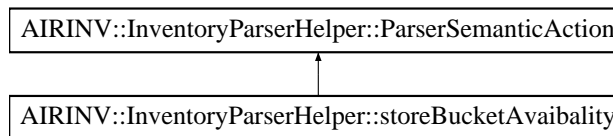
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.118 AIRINV::InventoryParserHelper::storeBucketAvaibility Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeBucketAvaibility::



## Public Member Functions

- [storeBucketAvaibility](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) (double iReal) const

## Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

### 10.118.1 Detailed Description

Store the parsed bucket availability.

Definition at line 229 of file InventoryParserHelper.hpp.

### 10.118.2 Constructor & Destructor Documentation

#### 10.118.2.1 AIRINV::InventoryParserHelper::storeBucketAvaibility::storeBucketAvaibility ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 392 of file InventoryParserHelper.cpp.

### 10.118.3 Member Function Documentation

#### 10.118.3.1 void AIRINV::InventoryParserHelper::storeBucketAvaibility::operator() (double *i-Real*) const

Actor Function (functor).

Definition at line 397 of file InventoryParserHelper.cpp.

References [AIRINV::BucketStruct::\\_availability](#), [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), and [AIRINV::FlightDateStruct::\\_itBucket](#).

### 10.118.4 Member Data Documentation

#### 10.118.4.1 [FlightDateStruct&](#) [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#) [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by [AIRINV::InventoryParserHelper::doEndFlightDate::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFClasses::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFamilyCode::operator\(\)](#), and [AIRINV::InventoryParserHelper::storeRevenueAvailability::operator\(\)](#).

AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

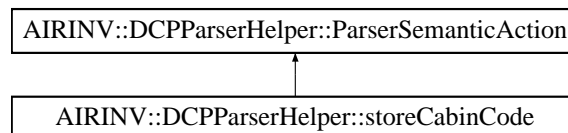
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.119 AIRINV::DCPParserHelper::storeCabinCode Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::storeCabinCode:



### Public Member Functions

- [storeCabinCode](#) (DCPRuleStruct &)

- void [operator\(\)](#) (char, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

## Public Attributes

- [DCPRuleStruct](#) & [\\_DCPRule](#)

### 10.119.1 Detailed Description

Store the cabin code.

Definition at line 118 of file [DCPParserHelper.hpp](#).

### 10.119.2 Constructor & Destructor Documentation

#### 10.119.2.1 AIRINV::DCPParserHelper::storeCabinCode::storeCabinCode (DCPRuleStruct &)

Actor Constructor.

Definition at line 166 of file [DCPParserHelper.cpp](#).

### 10.119.3 Member Function Documentation

#### 10.119.3.1 void AIRINV::DCPParserHelper::storeCabinCode::operator() (char, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

Actor Function (functor).

Definition at line 171 of file [DCPParserHelper.cpp](#).

References [AIRINV::DCPParserHelper::ParserSemanticAction::\\_DCPRule](#).

### 10.119.4 Member Data Documentation

#### 10.119.4.1 [DCPRuleStruct&](#) [AIRINV::DCPParserHelper::ParserSemanticAction::\\_DCPRule](#) [inherited]

Actor Context.

Definition at line 34 of file [DCPParserHelper.hpp](#).

Referenced by [AIRINV::DCPParserHelper::doEndDCP::operator\(\)](#), [AIRINV::DCPParserHelper::storeClass::operator\(\)](#), [AIRINV::DCPParserHelper::storeAirlineCode::operator\(\)](#), [AIRINV::DCPParserHelper::storeDCP::operator\(\)](#), [AIRINV::DCPParserHelper::storeMinimumStay::operator\(\)](#), [AIRINV::DCPParserHelper::storeNonRefundable::operator\(\)](#), [AIRINV::DCPParserHelper::storeChangeFees::operator\(\)](#), [AIRINV::DCPParserHelper::storeSaturdayStay::operator\(\)](#), [AIRINV::DCPParserHelper::storeAdvancePurchase::operator\(\)](#), [AIRINV::DCPParserHelper::storeChannel::operator\(\)](#), [operator\(\)](#), [AIRINV::DCPParserHelper::storePOS::operator\(\)](#), [AIRINV::DCPParserHelper::storeEndRangeTime::operator\(\)](#), [AIRINV::DCPParserHelper::storeStartRangeTime::operator\(\)](#), [AIRINV::DCPParserHelper::storeDateRangeEnd::operator\(\)](#), [AIRINV::DCPParserHelper::storeDateRangeStart::operator\(\)](#), [AIRINV::DCPParserHelper::storeDestination::operator\(\)](#), [AIRINV::DCPParserHelper::storeOrigin::operator\(\)](#), and [AIRINV::DCPParserHelper::storeDCPId::operator\(\)](#).

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

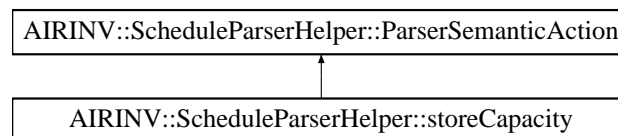
- [airinv/command/vault/DCPParserHelper.hpp](#)

- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.120 AIRINV::ScheduleParserHelper::storeCapacity Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeCapacity::



### Public Member Functions

- [storeCapacity](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) (double *iReal*) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.120.1 Detailed Description

Store the parsed capacity.

Definition at line 141 of file [ScheduleParserHelper.hpp](#).

#### 10.120.2 Constructor & Destructor Documentation

##### 10.120.2.1 AIRINV::ScheduleParserHelper::storeCapacity::storeCapacity ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 262 of file [ScheduleParserHelper.cpp](#).

#### 10.120.3 Member Function Documentation

##### 10.120.3.1 void AIRINV::ScheduleParserHelper::storeCapacity::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 267 of file [ScheduleParserHelper.cpp](#).

References [AIRINV::LegStruct::\\_cabinList](#), [AIRINV::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#), [AIRINV::FlightPeriodStruct::\\_itLeg](#), [AIRINV::FlightPeriodStruct::\\_itLegCabin](#), and [AIRINV::LegCabinStruct::\\_saleableCapacity](#).

#### 10.120.4 Member Data Documentation

##### 10.120.4.1 FlightPeriodStruct& AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flight-Period [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

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

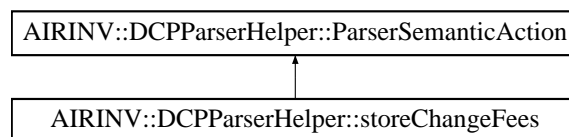
The documentation for this struct was generated from the following files:

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

## 10.121 AIRINV::DCPParserHelper::storeChangeFees Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::storeChangeFees::



### Public Member Functions

- [storeChangeFees](#) (DCPRuleStruct &)
- [operator\(\)](#) (char, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

### Public Attributes

- [DCPRuleStruct](#) & [\\_DCPRule](#)

### 10.121.1 Detailed Description

Store the parsed change fees.

Definition at line 158 of file DCPParserHelper.hpp.

### 10.121.2 Constructor & Destructor Documentation

#### 10.121.2.1 AIRINV::DCPParserHelper::storeChangeFees::storeChangeFees (DCPRuleStruct &)

Actor Constructor.

Definition at line 248 of file DCPParserHelper.cpp.

### 10.121.3 Member Function Documentation

#### 10.121.3.1 void AIRINV::DCPParserHelper::storeChangeFees::operator() (char, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

Actor Function (functor).

Definition at line 253 of file DCPParserHelper.cpp.

References AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule.

### 10.121.4 Member Data Documentation

#### 10.121.4.1 DCPRuleStruct& AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule [inherited]

Actor Context.

Definition at line 34 of file DCPParserHelper.hpp.

Referenced by AIRINV::DCPParserHelper::doEndDCP::operator(), AIRINV::DCPParserHelper::storeClass::operator(), AIRINV::DCPParserHelper::storeAirlineCode::operator(), AIRINV::DCPParserHelper::storeDCP::operator(), AIRINV::DCPParserHelper::storeMinimumStay::operator(), AIRINV::DCPParserHelper::storeNonRefundable::operator(), AIRINV::DCPParserHelper::storeSaturdayStay::operator(), AIRINV::DCPParserHelper::storeAdvancePurchase::operator(), AIRINV::DCPParserHelper::storeChannel::operator(), AIRINV::DCPParserHelper::storeCabinCode::operator(), AIRINV::DCPParserHelper::storePOS::operator(), AIRINV::DCPParserHelper::storeEndRangeTime::operator(), AIRINV::DCPParserHelper::storeStartRangeTime::operator(), AIRINV::DCPParserHelper::storeDateRangeEnd::operator(), AIRINV::DCPParserHelper::storeDateRangeStart::operator(), AIRINV::DCPParserHelper::storeDestination::operator(), AIRINV::DCPParserHelper::storeOrigin::operator(), and AIRINV::DCPParserHelper::storeDCPID::operator().

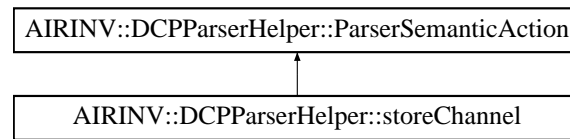
The documentation for this struct was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.122 AIRINV::DCPParserHelper::storeChannel Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::storeChannel::



## Public Member Functions

- [storeChannel](#) (DCPRuleStruct &)
- void [operator\(\)](#) (std::vector< char >, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

## Public Attributes

- DCPRuleStruct & [\\_DCPRule](#)

### 10.122.1 Detailed Description

Store the channel distribution.

Definition at line 128 of file DCPParserHelper.hpp.

### 10.122.2 Constructor & Destructor Documentation

#### 10.122.2.1 AIRINV::DCPParserHelper::storeChannel::storeChannel (DCPRuleStruct &)

Actor Constructor.

Definition at line 187 of file DCPParserHelper.cpp.

### 10.122.3 Member Function Documentation

#### 10.122.3.1 void AIRINV::DCPParserHelper::storeChannel::operator() (std::vector< char >, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

Actor Function (functor).

Definition at line 192 of file DCPParserHelper.cpp.

References AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule.

### 10.122.4 Member Data Documentation

#### 10.122.4.1 DCPRuleStruct& [AIRINV::DCPParserHelper::ParserSemanticAction::\\_DCPRule](#) [inherited]

Actor Context.

Definition at line 34 of file DCPParserHelper.hpp.

Referenced by AIRINV::DCPParserHelper::doEndDCP::operator>(), AIRINV::DCPParserHelper::storeClass::operator>(), AIRINV::DCPParserHelper::storeAirlineCode::operator>(),



AIRINV::DCPParserHelper::storeDCP::operator>(), AIRINV::DCPParserHelper::storeMinimumStay::operator>(), AIRINV::DCPParserHelper::storeNonRefundable::operator>(), AIRINV::DCPParserHelper::storeChangeFees::operator>(), AIRINV::DCPParserHelper::storeSaturdayStay::operator>(), AIRINV::DCPParserHelper::storeAdvancePurchase::operator>(), AIRINV::DCPParserHelper::storeCabinCode::operator>(), AIRINV::DCPParserHelper::storePOS::operator>(), AIRINV::DCPParserHelper::storeEndRangeTime::operator>(), AIRINV::DCPParserHelper::storeStartRangeTime::operator>(), AIRINV::DCPParserHelper::storeDateRangeEnd::operator>(), AIRINV::DCPParserHelper::storeDateRangeStart::operator>(), AIRINV::DCPParserHelper::storeDestination::operator>(), AIRINV::DCPParserHelper::storeOrigin::operator>(), and AIRINV::DCPParserHelper::storeDCPID::operator()).

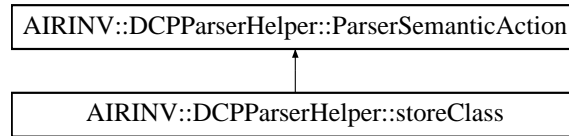
The documentation for this struct was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.123 AIRINV::DCPParserHelper::storeClass Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::storeClass::



### Public Member Functions

- [storeClass](#) (DCPRuleStruct &)
- void [operator\(\)](#) (std::vector< char >, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

### Public Attributes

- DCPRuleStruct & [\\_DCPRule](#)

#### 10.123.1 Detailed Description

Store the parsed class.

Definition at line 208 of file DCPParserHelper.hpp.

#### 10.123.2 Constructor & Destructor Documentation

##### 10.123.2.1 AIRINV::DCPParserHelper::storeClass::storeClass (DCPRuleStruct &)

Actor Constructor.

Definition at line 376 of file DCPParserHelper.cpp.

### 10.123.3 Member Function Documentation

#### 10.123.3.1 void AIRINV::DCPParserHelper::storeClass::operator() (std::vector< char >, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

Actor Function (functor).

Definition at line 381 of file DCPParserHelper.cpp.

References AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule.

### 10.123.4 Member Data Documentation

#### 10.123.4.1 DCPRuleStruct& AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule [inherited]

Actor Context.

Definition at line 34 of file DCPParserHelper.hpp.

Referenced by AIRINV::DCPParserHelper::doEndDCP::operator()(), operator()(), AIRINV::DCPParserHelper::storeAirlineCode::operator()(), AIRINV::DCPParserHelper::storeDCP::operator()(), AIRINV::DCPParserHelper::storeMinimumStay::operator()(), AIRINV::DCPParserHelper::storeNonRefundable::operator()(), AIRINV::DCPParserHelper::storeChangeFees::operator()(), AIRINV::DCPParserHelper::storeSaturdayStay::operator()(), AIRINV::DCPParserHelper::storeAdvancePurchase::operator()(), AIRINV::DCPParserHelper::storeChannel::operator()(), AIRINV::DCPParserHelper::storeCabinCode::operator()(), AIRINV::DCPParserHelper::storePOS::operator()(), AIRINV::DCPParserHelper::storeEndRangeTime::operator()(), AIRINV::DCPParserHelper::storeStartRangeTime::operator()(), AIRINV::DCPParserHelper::storeDateRangeEnd::operator()(), AIRINV::DCPParserHelper::storeDateRangeStart::operator()(), AIRINV::DCPParserHelper::storeDestination::operator()(), AIRINV::DCPParserHelper::storeOrigin::operator()(), and AIRINV::DCPParserHelper::storeDCPID::operator()().

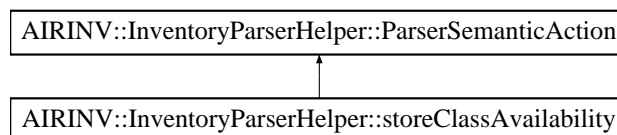
The documentation for this struct was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.124 AIRINV::InventoryParserHelper::storeClassAvailability Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeClassAvailability:



### Public Member Functions

- [storeClassAvailability](#) ([FlightDateStruct](#) &)
- void [operator\(\)](#) (double iReal) const

**Public Attributes**

- [FlightDateStruct](#) & [\\_flightDate](#)

**10.124.1 Detailed Description**

Store the parsed number of net class availability (at booking class level).

Definition at line 399 of file InventoryParserHelper.hpp.

**10.124.2 Constructor & Destructor Documentation****10.124.2.1 AIRINV::InventoryParserHelper::storeClassAvailability::storeClassAvailability([FlightDateStruct](#) &)**

Actor Constructor.

Definition at line 702 of file InventoryParserHelper.cpp.

**10.124.3 Member Function Documentation****10.124.3.1 void AIRINV::InventoryParserHelper::storeClassAvailability::operator() (double *i-Real*) const**

Actor Function (functor).

Definition at line 707 of file InventoryParserHelper.cpp.

References [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), [AIRINV::FlightDateStruct::\\_itBookingClass](#), and [AIRINV::BookingClassStruct::\\_netClassAvailability](#).

**10.124.4 Member Data Documentation****10.124.4.1 [FlightDateStruct](#)& [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#) [inherited]**

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by [AIRINV::InventoryParserHelper::doEndFlightDate::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFClasses::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFamilyCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeRevenueAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeSegmentAvailability::operator\(\)](#), [operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassETB::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeOverbooking::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNoShow::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNego::operator\(\)](#), [AIRINV::InventoryParserHelper::storeProtection::operator\(\)](#), [AIRINV::InventoryParserHelper::storeCumulatedProtection::operator\(\)](#), [AIRINV::InventoryParserHelper::storeParentSubclassCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeParentClassCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeSubclassCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator\(\)](#), and [AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator\(\)](#).

AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

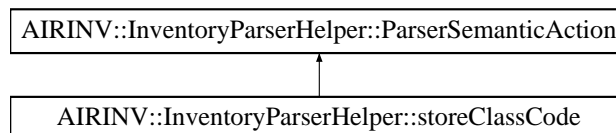
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.125 AIRINV::InventoryParserHelper::storeClassCode Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeClassCode::



### Public Member Functions

- [storeClassCode](#) ([FlightDateStruct](#) &)
- [operator\(\)](#) (char iChar) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.125.1 Detailed Description

Store the parsed booking class code.

Definition at line 277 of file [InventoryParserHelper.hpp](#).

## 10.125.2 Constructor & Destructor Documentation

### 10.125.2.1 AIRINV::InventoryParserHelper::storeClassCode::storeClassCode ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 524 of file InventoryParserHelper.cpp.

## 10.125.3 Member Function Documentation

### 10.125.3.1 void AIRINV::InventoryParserHelper::storeClassCode::operator() (char *iChar*) const

Actor Function (functor).

Definition at line 529 of file InventoryParserHelper.cpp.

References AIRINV::BookingClassStruct::\_classCode, AIRINV::FareFamilyStruct::\_classList, AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_itBookingClass, AIRINV::SegmentCabinStruct::\_itFareFamily, and AIRINV::FlightDateStruct::\_itSegmentCabin.

## 10.125.4 Member Data Documentation

### 10.125.4.1 [FlightDateStruct](#) & AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::store-

AU::operator>(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator>(),  
 AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParser-  
 Helper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(),  
 AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParser-  
 Helper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlight-  
 Number::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(),  
 AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::store-  
 LegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(),  
 AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParser-  
 Helper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(),  
 AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParser-  
 Helper::storeSnapshotDate::operator().

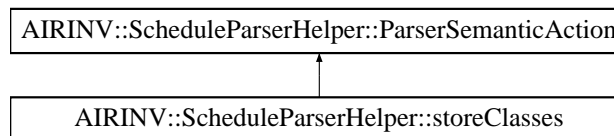
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.126 AIRINV::ScheduleParserHelper::storeClasses Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeClasses::



### Public Member Functions

- [storeClasses](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.126.1 Detailed Description

Store the parsed list of class codes.

Definition at line 184 of file [ScheduleParserHelper.hpp](#).

#### 10.126.2 Constructor & Destructor Documentation

##### 10.126.2.1 AIRINV::ScheduleParserHelper::storeClasses::storeClasses ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 344 of file [ScheduleParserHelper.cpp](#).

### 10.126.3 Member Function Documentation

#### 10.126.3.1 void AIRINV::ScheduleParserHelper::storeClasses::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 349 of file ScheduleParserHelper.cpp.

References AIRINV::FlightPeriodStruct::\_areSegmentDefinitionsSpecific, AIRINV::FareFamilyStruct::\_classes, AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, AIRINV::SegmentCabinStruct::\_itFareFamily, AIRINV::FlightPeriodStruct::\_itSegment, AIRINV::FlightPeriodStruct::\_itSegmentCabin, and AIRINV::FlightPeriodStruct::addSegmentCabin().

### 10.126.4 Member Data Documentation

#### 10.126.4.1 FlightPeriodStruct& AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

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

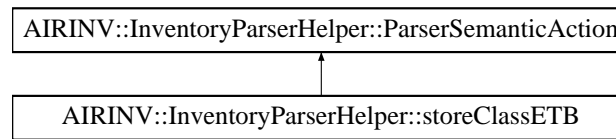
The documentation for this struct was generated from the following files:

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

## 10.127 AIRINV::InventoryParserHelper::storeClassETB Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeClassETB::



## Public Member Functions

- [storeClassETB](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) (double iReal) const

## Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

### 10.127.1 Detailed Description

Store the parsed expected to board number (at booking class level).

Definition at line 390 of file InventoryParserHelper.hpp.

### 10.127.2 Constructor & Destructor Documentation

#### 10.127.2.1 AIRINV::InventoryParserHelper::storeClassETB::storeClassETB ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 690 of file InventoryParserHelper.cpp.

### 10.127.3 Member Function Documentation

#### 10.127.3.1 void AIRINV::InventoryParserHelper::storeClassETB::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 695 of file InventoryParserHelper.cpp.

References [AIRINV::BookingClassStruct::\\_etb](#), [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), and [AIRINV::FlightDateStruct::\\_itBookingClass](#).

### 10.127.4 Member Data Documentation

#### 10.127.4.1 [FlightDateStruct&](#) [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#) [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by [AIRINV::InventoryParserHelper::doEndFlightDate::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFClasses::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFamilyCode::operator\(\)](#), and [AIRINV::InventoryParserHelper::storeRevenueAvailability::operator\(\)](#).



AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

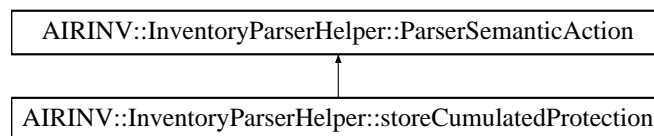
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.128 AIRINV::InventoryParserHelper::storeCumulatedProtection Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeCumulatedProtection::



## Public Member Functions

- [storeCumulatedProtection](#) ([FlightDateStruct](#) &)
- void [operator\(\)](#) (double iReal) const

## Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

### 10.128.1 Detailed Description

Store the parsed cumulated protection (at booking class level).

Definition at line 309 of file InventoryParserHelper.hpp.

### 10.128.2 Constructor & Destructor Documentation

#### 10.128.2.1 AIRINV::InventoryParserHelper::storeCumulatedProtection::storeCumulatedProtection ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 579 of file InventoryParserHelper.cpp.

### 10.128.3 Member Function Documentation

#### 10.128.3.1 void AIRINV::InventoryParserHelper::storeCumulatedProtection::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 584 of file InventoryParserHelper.cpp.

References [AIRINV::BookingClassStruct::\\_cumulatedProtection](#), [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), and [AIRINV::FlightDateStruct::\\_itBookingClass](#).

### 10.128.4 Member Data Documentation

#### 10.128.4.1 [FlightDateStruct&](#) [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#) [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by [AIRINV::InventoryParserHelper::doEndFlightDate::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFClasses::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFamilyCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeRevenueAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeSegmentAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassETB::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator\(\)](#), and [AIRINV::InventoryParserHelper::storeNbOfBkgs::operator\(\)](#).

AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

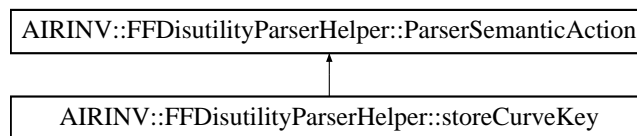
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.129 AIRINV::FFDisutilityParserHelper::storeCurveKey Struct Reference

```
#include <airinv/command/FFDisutilityParserHelper.hpp>
```

Inheritance diagram for AIRINV::FFDisutilityParserHelper::storeCurveKey::



### Public Member Functions

- [storeCurveKey](#) (FFDisutilityStruct &)
- [void operator\(\)](#) (iterator\_t iStr, iterator\_t iStrEnd) const

**Public Attributes**

- [FFDisutilityStruct](#) & [\\_ffDisutility](#)

**10.129.1 Detailed Description**

Store the parsed curve key.

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

**10.129.2 Constructor & Destructor Documentation****10.129.2.1 AIRINV::FFDisutilityParserHelper::storeCurveKey::storeCurveKey ([FFDisutilityStruct](#) &)**

Actor Constructor.

Definition at line 33 of file [FFDisutilityParserHelper.cpp](#).

**10.129.3 Member Function Documentation****10.129.3.1 void AIRINV::FFDisutilityParserHelper::storeCurveKey::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const**

Actor Function (functor).

Definition at line 38 of file [FFDisutilityParserHelper.cpp](#).

References [AIRINV::FFDisutilityParserHelper::ParserSemanticAction::\\_ffDisutility](#), and [AIRINV::FFDisutilityStruct::\\_key](#).

**10.129.4 Member Data Documentation****10.129.4.1 [FFDisutilityStruct](#)& AIRINV::FFDisutilityParserHelper::ParserSemanticAction::\_ffDisutility [inherited]**

Actor Context.

Definition at line 33 of file [FFDisutilityParserHelper.hpp](#).

Referenced by [AIRINV::FFDisutilityParserHelper::doEndCurve::operator\(\)\(\)](#), [AIRINV::FFDisutilityParserHelper::storeFFDisutilityValue::operator\(\)\(\)](#), [AIRINV::FFDisutilityParserHelper::storeDTD::operator\(\)\(\)](#), and [operator\(\)\(\)](#).

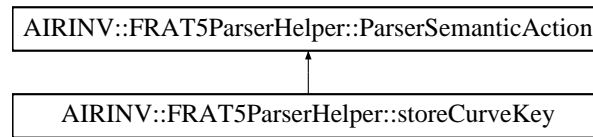
The documentation for this struct was generated from the following files:

- [airinv/command/FFDisutilityParserHelper.hpp](#)
- [airinv/command/FFDisutilityParserHelper.cpp](#)

**10.130 AIRINV::FRAT5ParserHelper::storeCurveKey Struct Reference**

```
#include <airinv/command/FRAT5ParserHelper.hpp>
```

Inheritance diagram for AIRINV::FRAT5ParserHelper::storeCurveKey::



## Public Member Functions

- [storeCurveKey](#) ([FRAT5Struct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

## Public Attributes

- [FRAT5Struct](#) & [\\_frat5](#)

### 10.130.1 Detailed Description

Store the parsed curve key.

Definition at line 37 of file `FRAT5ParserHelper.hpp`.

### 10.130.2 Constructor & Destructor Documentation

#### 10.130.2.1 AIRINV::FRAT5ParserHelper::storeCurveKey::storeCurveKey ([FRAT5Struct](#) &)

Actor Constructor.

Definition at line 33 of file `FRAT5ParserHelper.cpp`.

### 10.130.3 Member Function Documentation

#### 10.130.3.1 void AIRINV::FRAT5ParserHelper::storeCurveKey::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 38 of file `FRAT5ParserHelper.cpp`.

References `AIRINV::FRAT5ParserHelper::ParserSemanticAction::_frat5`, and `AIRINV::FRAT5Struct::_key`.

### 10.130.4 Member Data Documentation

#### 10.130.4.1 [FRAT5Struct](#)& [AIRINV::FRAT5ParserHelper::ParserSemanticAction::\\_frat5](#) [inherited]

Actor Context.

Definition at line 33 of file `FRAT5ParserHelper.hpp`.

Referenced by `AIRINV::FRAT5ParserHelper::doEndCurve::operator()`, `AIRINV::FRAT5ParserHelper::storeFRAT5Value::operator()`, `AIRINV::FRAT5ParserHelper::storeDTD::operator()`, and `operator()`.

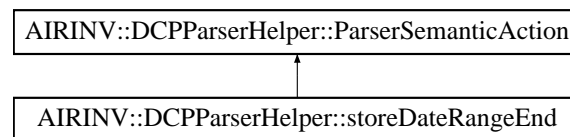
The documentation for this struct was generated from the following files:

- [airinv/command/FRAT5ParserHelper.hpp](#)
- [airinv/command/FRAT5ParserHelper.cpp](#)

## 10.131 AIRINV::DCPParserHelper::storeDateRangeEnd Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::storeDateRangeEnd::



### Public Member Functions

- [storeDateRangeEnd](#) (DCPRuleStruct &)
- void [operator\(\)](#) (boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

### Public Attributes

- DCPRuleStruct & [\\_DCPRule](#)

#### 10.131.1 Detailed Description

Store the parsed end of the date range.

Definition at line 78 of file DCPParserHelper.hpp.

#### 10.131.2 Constructor & Destructor Documentation

##### 10.131.2.1 AIRINV::DCPParserHelper::storeDateRangeEnd::storeDateRangeEnd (DCPRuleStruct &)

Actor Constructor.

Definition at line 101 of file DCPParserHelper.cpp.

#### 10.131.3 Member Function Documentation

##### 10.131.3.1 void AIRINV::DCPParserHelper::storeDateRangeEnd::operator() (boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

Actor Function (functor).

Definition at line 106 of file DCPParserHelper.cpp.

References AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule.

### 10.131.4 Member Data Documentation

#### 10.131.4.1 DCPRuleStruct& AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule [inherited]

Actor Context.

Definition at line 34 of file DCPParserHelper.hpp.

Referenced by AIRINV::DCPParserHelper::doEndDCP::operator(), AIRINV::DCPParserHelper::storeClass::operator(), AIRINV::DCPParserHelper::storeAirlineCode::operator(), AIRINV::DCPParserHelper::storeDCP::operator(), AIRINV::DCPParserHelper::storeMinimumStay::operator(), AIRINV::DCPParserHelper::storeNonRefundable::operator(), AIRINV::DCPParserHelper::storeChangeFees::operator(), AIRINV::DCPParserHelper::storeSaturdayStay::operator(), AIRINV::DCPParserHelper::storeAdvancePurchase::operator(), AIRINV::DCPParserHelper::storeChannel::operator(), AIRINV::DCPParserHelper::storeCabinCode::operator(), AIRINV::DCPParserHelper::storePOS::operator(), AIRINV::DCPParserHelper::storeEndRangeTime::operator(), AIRINV::DCPParserHelper::storeStartRangeTime::operator(), operator(), AIRINV::DCPParserHelper::storeDateRangeStart::operator(), AIRINV::DCPParserHelper::storeDestination::operator(), AIRINV::DCPParserHelper::storeOrigin::operator(), and AIRINV::DCPParserHelper::storeDCPId::operator().

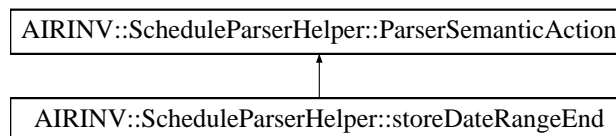
The documentation for this struct was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.132 AIRINV::ScheduleParserHelper::storeDateRangeEnd Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeDateRangeEnd::



### Public Member Functions

- [storeDateRangeEnd](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

### 10.132.1 Detailed Description

Store the end of the date range.

Definition at line 61 of file ScheduleParserHelper.hpp.

### 10.132.2 Constructor & Destructor Documentation

#### 10.132.2.1 AIRINV::ScheduleParserHelper::storeDateRangeEnd::storeDateRangeEnd ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 77 of file ScheduleParserHelper.cpp.

### 10.132.3 Member Function Documentation

#### 10.132.3.1 void AIRINV::ScheduleParserHelper::storeDateRangeEnd::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 82 of file ScheduleParserHelper.cpp.

References AIRINV::FlightPeriodStruct::\_airlineCode, AIRINV::LegStruct::\_airlineCode, AIRINV::FlightPeriodStruct::\_dateRange, AIRINV::FlightPeriodStruct::\_dateRangeEnd, AIRINV::FlightPeriodStruct::\_dateRangeStart, AIRINV::FlightPeriodStruct::\_flightNumber, AIRINV::LegStruct::\_flightNumber, AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, AIRINV::FlightPeriodStruct::\_itLeg, AIRINV::FlightPeriodStruct::\_itSeconds, and AIRINV::FlightPeriodStruct::getDate().

### 10.132.4 Member Data Documentation

#### 10.132.4.1 [FlightPeriodStruct&](#) AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

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

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

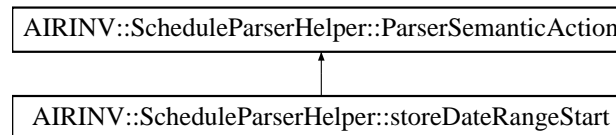
- [airinv/command/ScheduleParserHelper.hpp](#)
- [airinv/command/ScheduleParserHelper.cpp](#)



## 10.133 AIRINV::ScheduleParserHelper::storeDateRangeStart Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeDateRangeStart::



### Public Member Functions

- [storeDateRangeStart](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.133.1 Detailed Description

Store the start of the date range.

Definition at line 53 of file ScheduleParserHelper.hpp.

#### 10.133.2 Constructor & Destructor Documentation

##### 10.133.2.1 AIRINV::ScheduleParserHelper::storeDateRangeStart::storeDateRangeStart ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 62 of file ScheduleParserHelper.cpp.

#### 10.133.3 Member Function Documentation

##### 10.133.3.1 void AIRINV::ScheduleParserHelper::storeDateRangeStart::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 67 of file ScheduleParserHelper.cpp.

References [AIRINV::FlightPeriodStruct::\\_dateRangeStart](#), [AIRINV::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#), [AIRINV::FlightPeriodStruct::\\_itSeconds](#), and [AIRINV::FlightPeriodStruct::getDate\(\)](#).

#### 10.133.4 Member Data Documentation

##### 10.133.4.1 [FlightPeriodStruct](#)& [AIRINV::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#) [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

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

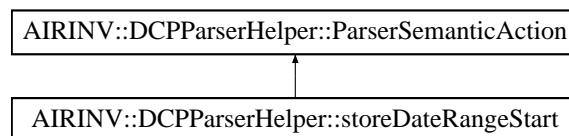
The documentation for this struct was generated from the following files:

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

## 10.134 AIRINV::DCPParserHelper::storeDateRangeStart Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::storeDateRangeStart::



### Public Member Functions

- [storeDateRangeStart](#) (DCPRuleStruct &)
- void [operator\(\)](#) (boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

### Public Attributes

- DCPRuleStruct & [\\_DCPRule](#)

#### 10.134.1 Detailed Description

Store the parsed start of the date range.

Definition at line 68 of file DCPParserHelper.hpp.

### 10.134.2 Constructor & Destructor Documentation

#### 10.134.2.1 AIRINV::DCPParserHelper::storeDateRangeStart::storeDateRangeStart (DCPRule-Struct &)

Actor Constructor.

Definition at line 86 of file DCPParserHelper.cpp.

### 10.134.3 Member Function Documentation

#### 10.134.3.1 void AIRINV::DCPParserHelper::storeDateRangeStart::operator() (boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

Actor Function (functor).

Definition at line 91 of file DCPParserHelper.cpp.

References AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule.

### 10.134.4 Member Data Documentation

#### 10.134.4.1 DCPRuleStruct& AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule [inherited]

Actor Context.

Definition at line 34 of file DCPParserHelper.hpp.

Referenced by AIRINV::DCPParserHelper::doEndDCP::operator(), AIRINV::DCPParserHelper::storeClass::operator(), AIRINV::DCPParserHelper::storeAirlineCode::operator(), AIRINV::DCPParserHelper::storeDCP::operator(), AIRINV::DCPParserHelper::storeMinimumStay::operator(), AIRINV::DCPParserHelper::storeNonRefundable::operator(), AIRINV::DCPParserHelper::storeChangeFees::operator(), AIRINV::DCPParserHelper::storeSaturdayStay::operator(), AIRINV::DCPParserHelper::storeAdvancePurchase::operator(), AIRINV::DCPParserHelper::storeChannel::operator(), AIRINV::DCPParserHelper::storeCabinCode::operator(), AIRINV::DCPParserHelper::storePOS::operator(), AIRINV::DCPParserHelper::storeEndRangeTime::operator(), AIRINV::DCPParserHelper::storeStartRangeTime::operator(), AIRINV::DCPParserHelper::storeDateRangeEnd::operator(), operator(), AIRINV::DCPParserHelper::storeDestination::operator(), AIRINV::DCPParserHelper::storeOrigin::operator(), and AIRINV::DCPParserHelper::storeDCPId::operator().

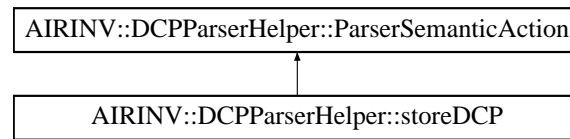
The documentation for this struct was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.135 AIRINV::DCPParserHelper::storeDCP Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::storeDCP::



## Public Member Functions

- [storeDCP](#) (DCPRuleStruct &)
- void [operator\(\)](#) (double, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

## Public Attributes

- DCPRuleStruct & [\\_DCPRule](#)

### 10.135.1 Detailed Description

Store the parsed DCP value.

Definition at line 188 of file DCPParserHelper.hpp.

### 10.135.2 Constructor & Destructor Documentation

#### 10.135.2.1 AIRINV::DCPParserHelper::storeDCP::storeDCP (DCPRuleStruct &)

Actor Constructor.

Definition at line 314 of file DCPParserHelper.cpp.

### 10.135.3 Member Function Documentation

#### 10.135.3.1 void AIRINV::DCPParserHelper::storeDCP::operator() (double, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

Actor Function (functor).

Definition at line 319 of file DCPParserHelper.cpp.

References AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule.

### 10.135.4 Member Data Documentation

#### 10.135.4.1 DCPRuleStruct& AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule [inherited]

Actor Context.

Definition at line 34 of file DCPParserHelper.hpp.

Referenced by AIRINV::DCPParserHelper::doEndDCP::operator(), AIRINV::DCPParserHelper::storeClass::operator(), AIRINV::DCPParserHelper::storeAirlineCode::operator(), operator(), AIRINV::DCPParserHelper::storeMinimumStay::operator(), AIRINV::DCPParserHelper::storeNonRefundable::operator(), AIRINV::DCPParserHelper::storeChangeFees::operator(),

AIRINV::DCPParserHelper::storeSaturdayStay::operator(), AIRINV::DCPParserHelper::storeAdvancePurchase::operator(), AIRINV::DCPParserHelper::storeChannel::operator(), AIRINV::DCPParserHelper::storeCabinCode::operator(), AIRINV::DCPParserHelper::storePOS::operator(), AIRINV::DCPParserHelper::storeEndRangeTime::operator(), AIRINV::DCPParserHelper::storeStartRangeTime::operator(), AIRINV::DCPParserHelper::storeDateRangeEnd::operator(), AIRINV::DCPParserHelper::storeDateRangeStart::operator(), AIRINV::DCPParserHelper::storeDestination::operator(), AIRINV::DCPParserHelper::storeOrigin::operator(), and AIRINV::DCPParserHelper::storeDCPId::operator().

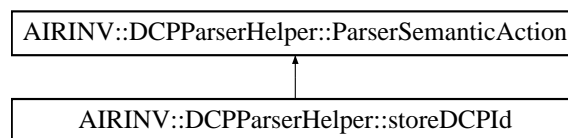
The documentation for this struct was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.136 AIRINV::DCPParserHelper::storeDCPId Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::storeDCPId::



### Public Member Functions

- [storeDCPId](#) (DCPRuleStruct &)
- void [operator\(\)](#) (unsigned int, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

### Public Attributes

- DCPRuleStruct & [\\_DCPRule](#)

#### 10.136.1 Detailed Description

Store the parsed DCP Id.

Definition at line 38 of file DCPParserHelper.hpp.

#### 10.136.2 Constructor & Destructor Documentation

##### 10.136.2.1 AIRINV::DCPParserHelper::storeDCPId::storeDCPId (DCPRuleStruct &)

Actor Constructor.

Definition at line 30 of file DCPParserHelper.cpp.

### 10.136.3 Member Function Documentation

#### 10.136.3.1 void AIRINV::DCPParserHelper::storeDCPID::operator() (unsigned int, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

Actor Function (functor).

Definition at line 35 of file DCPParserHelper.cpp.

References AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule.

### 10.136.4 Member Data Documentation

#### 10.136.4.1 DCPRuleStruct& AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule [inherited]

Actor Context.

Definition at line 34 of file DCPParserHelper.hpp.

Referenced by AIRINV::DCPParserHelper::doEndDCP::operator(), AIRINV::DCPParserHelper::storeClass::operator(), AIRINV::DCPParserHelper::storeAirlineCode::operator(), AIRINV::DCPParserHelper::storeDCP::operator(), AIRINV::DCPParserHelper::storeMinimumStay::operator(), AIRINV::DCPParserHelper::storeNonRefundable::operator(), AIRINV::DCPParserHelper::storeChangeFees::operator(), AIRINV::DCPParserHelper::storeSaturdayStay::operator(), AIRINV::DCPParserHelper::storeAdvancePurchase::operator(), AIRINV::DCPParserHelper::storeChannel::operator(), AIRINV::DCPParserHelper::storeCabinCode::operator(), AIRINV::DCPParserHelper::storePOS::operator(), AIRINV::DCPParserHelper::storeEndRangeTime::operator(), AIRINV::DCPParserHelper::storeStartRangeTime::operator(), AIRINV::DCPParserHelper::storeDateRangeEnd::operator(), AIRINV::DCPParserHelper::storeDateRangeStart::operator(), AIRINV::DCPParserHelper::storeDestination::operator(), AIRINV::DCPParserHelper::storeOrigin::operator(), and operator().

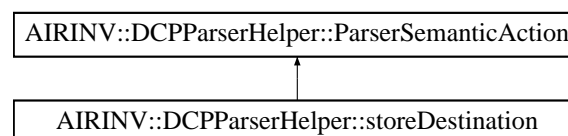
The documentation for this struct was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.137 AIRINV::DCPParserHelper::storeDestination Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::storeDestination::



### Public Member Functions

- [storeDestination](#) (DCPRuleStruct &)
- void [operator\(\)](#) (std::vector< char >, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

**Public Attributes**

- [DCPRuleStruct & \\_DCPRule](#)

**10.137.1 Detailed Description**

Store the parsed destination.

Definition at line 58 of file [DCPParserHelper.hpp](#).

**10.137.2 Constructor & Destructor Documentation****10.137.2.1 AIRINV::DCPParserHelper::storeDestination::storeDestination (DCPRuleStruct &)**

Actor Constructor.

Definition at line 70 of file [DCPParserHelper.cpp](#).

**10.137.3 Member Function Documentation****10.137.3.1 void AIRINV::DCPParserHelper::storeDestination::operator() (std::vector< char >, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const**

Actor Function (functor).

Definition at line 75 of file [DCPParserHelper.cpp](#).

References [AIRINV::DCPParserHelper::ParserSemanticAction::\\_DCPRule](#).

**10.137.4 Member Data Documentation****10.137.4.1 DCPRuleStruct& [AIRINV::DCPParserHelper::ParserSemanticAction::\\_DCPRule](#) [inherited]**

Actor Context.

Definition at line 34 of file [DCPParserHelper.hpp](#).

Referenced by [AIRINV::DCPParserHelper::doEndDCP::operator\(\)](#), [AIRINV::DCPParserHelper::storeClass::operator\(\)](#), [AIRINV::DCPParserHelper::storeAirlineCode::operator\(\)](#), [AIRINV::DCPParserHelper::storeDCP::operator\(\)](#), [AIRINV::DCPParserHelper::storeMinimumStay::operator\(\)](#), [AIRINV::DCPParserHelper::storeNonRefundable::operator\(\)](#), [AIRINV::DCPParserHelper::storeChangeFees::operator\(\)](#), [AIRINV::DCPParserHelper::storeSaturdayStay::operator\(\)](#), [AIRINV::DCPParserHelper::storeAdvancePurchase::operator\(\)](#), [AIRINV::DCPParserHelper::storeChannel::operator\(\)](#), [AIRINV::DCPParserHelper::storeCabinCode::operator\(\)](#), [AIRINV::DCPParserHelper::storePOS::operator\(\)](#), [AIRINV::DCPParserHelper::storeEndRangeTime::operator\(\)](#), [AIRINV::DCPParserHelper::storeStartRangeTime::operator\(\)](#), [AIRINV::DCPParserHelper::storeDateRangeEnd::operator\(\)](#), [AIRINV::DCPParserHelper::storeDateRangeStart::operator\(\)](#), [operator\(\)](#), [AIRINV::DCPParserHelper::storeOrigin::operator\(\)](#), and [AIRINV::DCPParserHelper::storeDCPId::operator\(\)](#).

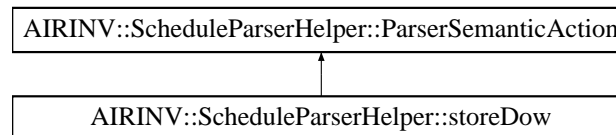
The documentation for this struct was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.138 AIRINV::ScheduleParserHelper::storeDow Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeDow::



### Public Member Functions

- [storeDow](#) ([FlightPeriodStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.138.1 Detailed Description

Store the DOW (day of the Week).

Definition at line 69 of file ScheduleParserHelper.hpp.

#### 10.138.2 Constructor & Destructor Documentation

##### 10.138.2.1 AIRINV::ScheduleParserHelper::storeDow::storeDow ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 105 of file ScheduleParserHelper.cpp.

#### 10.138.3 Member Function Documentation

##### 10.138.3.1 void AIRINV::ScheduleParserHelper::storeDow::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 110 of file ScheduleParserHelper.cpp.

References [AIRINV::FlightPeriodStruct::\\_dow](#), and [AIRINV::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#).

#### 10.138.4 Member Data Documentation

##### 10.138.4.1 [FlightPeriodStruct](#)& [AIRINV::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#) [inherited]

Actor Context.



Definition at line 33 of file ScheduleParserHelper.hpp.

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

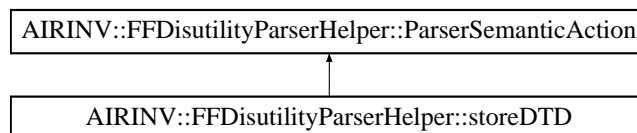
The documentation for this struct was generated from the following files:

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

## 10.139 AIRINV::FFDisutilityParserHelper::storeDTD Struct Reference

```
#include <airinv/command/FFDisutilityParserHelper.hpp>
```

Inheritance diagram for AIRINV::FFDisutilityParserHelper::storeDTD::



### Public Member Functions

- [storeDTD](#) ([FFDisutilityStruct](#) &)
- [void operator\(\)](#) (int iDTD) const

### Public Attributes

- [FFDisutilityStruct](#) & [\\_ffDisutility](#)

#### 10.139.1 Detailed Description

Store the DTD.

Definition at line 45 of file FFDisutilityParserHelper.hpp.

## 10.139.2 Constructor & Destructor Documentation

### 10.139.2.1 AIRINV::FFDisutilityParserHelper::storeDTD::storeDTD ([FFDisutilityStruct](#) &)

Actor Constructor.

Definition at line 45 of file `FFDisutilityParserHelper.cpp`.

## 10.139.3 Member Function Documentation

### 10.139.3.1 void AIRINV::FFDisutilityParserHelper::storeDTD::operator() (int *iDTD*) const

Actor Function (functor).

Definition at line 50 of file `FFDisutilityParserHelper.cpp`.

References `AIRINV::FFDisutilityStruct::_dtd`, and `AIRINV::FFDisutilityParserHelper::ParserSemanticAction::_ffDisutility`.

## 10.139.4 Member Data Documentation

### 10.139.4.1 [FFDisutilityStruct&](#) [AIRINV::FFDisutilityParserHelper::ParserSemanticAction::\\_ffDisutility](#) [inherited]

Actor Context.

Definition at line 33 of file `FFDisutilityParserHelper.hpp`.

Referenced by `AIRINV::FFDisutilityParserHelper::doEndCurve::operator()`, `AIRINV::FFDisutilityParserHelper::storeFFDisutilityValue::operator()`, `operator()`, and `AIRINV::FFDisutilityParserHelper::storeCurveKey::operator()`.

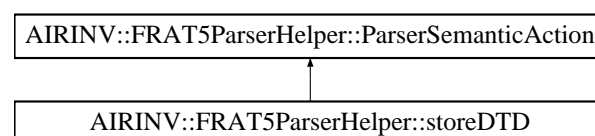
The documentation for this struct was generated from the following files:

- `airinv/command/FFDisutilityParserHelper.hpp`
- `airinv/command/FFDisutilityParserHelper.cpp`

## 10.140 AIRINV::FRAT5ParserHelper::storeDTD Struct Reference

```
#include <airinv/command/FRAT5ParserHelper.hpp>
```

Inheritance diagram for `AIRINV::FRAT5ParserHelper::storeDTD::`



## Public Member Functions

- [storeDTD](#) ([FRAT5Struct](#) &)
- void [operator\(\)](#) (int *iDTD*) const

**Public Attributes**

- [FRAT5Struct](#) & [\\_frat5](#)

**10.140.1 Detailed Description**

Store the DTD.

Definition at line 45 of file FRAT5ParserHelper.hpp.

**10.140.2 Constructor & Destructor Documentation****10.140.2.1 AIRINV::FRAT5ParserHelper::storeDTD::storeDTD ([FRAT5Struct](#) &)**

Actor Constructor.

Definition at line 46 of file FRAT5ParserHelper.cpp.

**10.140.3 Member Function Documentation****10.140.3.1 void AIRINV::FRAT5ParserHelper::storeDTD::operator() (int *iDTD*) const**

Actor Function (functor).

Definition at line 51 of file FRAT5ParserHelper.cpp.

References [AIRINV::FRAT5Struct::\\_dtd](#), and [AIRINV::FRAT5ParserHelper::ParserSemanticAction::\\_-frat5](#).

**10.140.4 Member Data Documentation****10.140.4.1 [FRAT5Struct](#)& [AIRINV::FRAT5ParserHelper::ParserSemanticAction::\\_frat5](#) [inherited]**

Actor Context.

Definition at line 33 of file FRAT5ParserHelper.hpp.

Referenced by [AIRINV::FRAT5ParserHelper::doEndCurve::operator\(\)](#), [AIRINV::FRAT5ParserHelper::storeFRAT5Value::operator\(\)](#), [operator\(\)](#), and [AIRINV::FRAT5ParserHelper::storeCurveKey::operator\(\)](#).

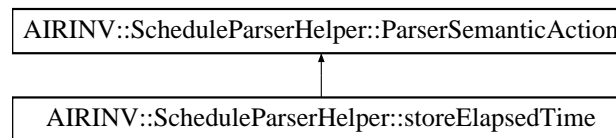
The documentation for this struct was generated from the following files:

- [airinv/command/FRAT5ParserHelper.hpp](#)
- [airinv/command/FRAT5ParserHelper.cpp](#)

**10.141 AIRINV::ScheduleParserHelper::storeElapsedTime Struct Reference**

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeElapsedTime::



## Public Member Functions

- [storeElapsedTime](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

## Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

### 10.141.1 Detailed Description

Store the elapsed time.

Definition at line 125 of file `ScheduleParserHelper.hpp`.

### 10.141.2 Constructor & Destructor Documentation

#### 10.141.2.1 AIRINV::ScheduleParserHelper::storeElapsedTime::storeElapsedTime ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 229 of file `ScheduleParserHelper.cpp`.

### 10.141.3 Member Function Documentation

#### 10.141.3.1 void AIRINV::ScheduleParserHelper::storeElapsedTime::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 234 of file `ScheduleParserHelper.cpp`.

References `AIRINV::FlightPeriodStruct::_dateOffset`, `AIRINV::LegStruct::_elapsed`, `AIRINV::ScheduleParserHelper::ParserSemanticAction::_flightPeriod`, `AIRINV::FlightPeriodStruct::_itLeg`, `AIRINV::FlightPeriodStruct::_itSeconds`, `AIRINV::LegStruct::_offDateOffset`, and `AIRINV::FlightPeriodStruct::getTime()`.

### 10.141.4 Member Data Documentation

#### 10.141.4.1 [FlightPeriodStruct](#)& AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [inherited]

Actor Context.

Definition at line 33 of file `ScheduleParserHelper.hpp`.

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

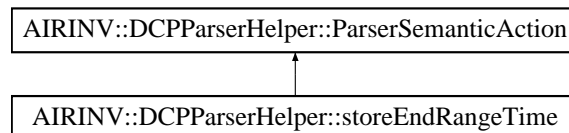
The documentation for this struct was generated from the following files:

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

## 10.142 AIRINV::DCPParserHelper::storeEndRangeTime Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::storeEndRangeTime::



### Public Member Functions

- [storeEndRangeTime](#) (DCPRuleStruct &)
- void [operator\(\)](#) (boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

### Public Attributes

- DCPRuleStruct & [\\_DCPRule](#)

#### 10.142.1 Detailed Description

Store the parsed end start range time.

Definition at line 98 of file DCPParserHelper.hpp.

## 10.142.2 Constructor & Destructor Documentation

### 10.142.2.1 AIRINV::DCPParserHelper::storeEndRangeTime::storeEndRangeTime (DCPRuleStruct &)

Actor Constructor.

Definition at line 133 of file DCPParserHelper.cpp.

## 10.142.3 Member Function Documentation

### 10.142.3.1 void AIRINV::DCPParserHelper::storeEndRangeTime::operator()(boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

Actor Function (functor).

Definition at line 138 of file DCPParserHelper.cpp.

References AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule.

## 10.142.4 Member Data Documentation

### 10.142.4.1 DCPRuleStruct& AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule [inherited]

Actor Context.

Definition at line 34 of file DCPParserHelper.hpp.

Referenced by AIRINV::DCPParserHelper::doEndDCP::operator(), AIRINV::DCPParserHelper::storeClass::operator(), AIRINV::DCPParserHelper::storeAirlineCode::operator(), AIRINV::DCPParserHelper::storeDCP::operator(), AIRINV::DCPParserHelper::storeMinimumStay::operator(), AIRINV::DCPParserHelper::storeNonRefundable::operator(), AIRINV::DCPParserHelper::storeChangeFees::operator(), AIRINV::DCPParserHelper::storeSaturdayStay::operator(), AIRINV::DCPParserHelper::storeAdvancePurchase::operator(), AIRINV::DCPParserHelper::storeChannel::operator(), AIRINV::DCPParserHelper::storeCabinCode::operator(), AIRINV::DCPParserHelper::storePOS::operator(), operator(), AIRINV::DCPParserHelper::storeStartRangeTime::operator(), AIRINV::DCPParserHelper::storeDateRangeEnd::operator(), AIRINV::DCPParserHelper::storeDateRangeStart::operator(), AIRINV::DCPParserHelper::storeDestination::operator(), AIRINV::DCPParserHelper::storeOrigin::operator(), and AIRINV::DCPParserHelper::storeDCPid::operator().

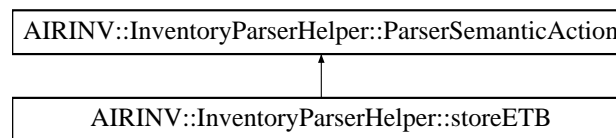
The documentation for this struct was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.143 AIRINV::InventoryParserHelper::storeETB Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeETB::



## Public Member Functions

- [storeETB](#) ([FlightDateStruct](#) &)
- void [operator\(\)](#) (double *iReal*) const

## Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

### 10.143.1 Detailed Description

Store the parsed Expected To Board (ETB) number.

Definition at line 213 of file `InventoryParserHelper.hpp`.

### 10.143.2 Constructor & Destructor Documentation

#### 10.143.2.1 AIRINV::InventoryParserHelper::storeETB::storeETB ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 361 of file `InventoryParserHelper.cpp`.

### 10.143.3 Member Function Documentation

#### 10.143.3.1 void AIRINV::InventoryParserHelper::storeETB::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 366 of file `InventoryParserHelper.cpp`.

References `AIRINV::LegCabinStruct::_etb`, `AIRINV::InventoryParserHelper::ParserSemanticAction::_flightDate`, and `AIRINV::FlightDateStruct::_itLegCabin`.

### 10.143.4 Member Data Documentation

#### 10.143.4.1 [FlightDateStruct](#)& [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#) [inherited]

Actor Context.

Definition at line 33 of file `InventoryParserHelper.hpp`.

Referenced by `AIRINV::InventoryParserHelper::doEndFlightDate::operator()`, `AIRINV::InventoryParserHelper::storeFClasses::operator()`, `AIRINV::InventoryParserHelper::storeFamilyCode::operator()`, and `AIRINV::InventoryParserHelper::storeRevenueAvailability::operator()`.

AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNegotiation::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

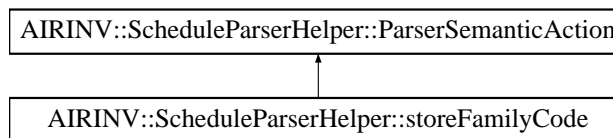
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.144 AIRINV::ScheduleParserHelper::storeFamilyCode Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeFamilyCode::





**Public Member Functions**

- [storeFamilyCode](#) ([FlightPeriodStruct](#) &)
- void [operator\(\)](#) (int iCode) const

**Public Attributes**

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

**10.144.1 Detailed Description**

Store the parsed family code.

Definition at line 192 of file `ScheduleParserHelper.hpp`.

**10.144.2 Constructor & Destructor Documentation****10.144.2.1 AIRINV::ScheduleParserHelper::storeFamilyCode::storeFamilyCode ([FlightPeriodStruct](#) &)**

Actor Constructor.

Definition at line 369 of file `ScheduleParserHelper.cpp`.

**10.144.3 Member Function Documentation****10.144.3.1 void AIRINV::ScheduleParserHelper::storeFamilyCode::operator() (int iCode) const**

Actor Function (functor).

Definition at line 374 of file `ScheduleParserHelper.cpp`.

References `AIRINV::FareFamilyStruct::_familyCode`, `AIRINV::ScheduleParserHelper::ParserSemanticAction::_flightPeriod`, `AIRINV::SegmentCabinStruct::_itFareFamily`, and `AIRINV::FlightPeriodStruct::_itSegmentCabin`.

**10.144.4 Member Data Documentation****10.144.4.1 [FlightPeriodStruct](#)& AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod** [*inherited*]

Actor Context.

Definition at line 33 of file `ScheduleParserHelper.hpp`.

Referenced by `AIRINV::ScheduleParserHelper::doEndFlight::operator()`, `AIRINV::ScheduleParserHelper::storeFClasses::operator()`, `AIRINV::ScheduleParserHelper::storeFFDisutilityCurveKey::operator()`, `AIRINV::ScheduleParserHelper::storeFRAT5CurveKey::operator()`, `operator()`, `AIRINV::ScheduleParserHelper::storeClasses::operator()`, `AIRINV::ScheduleParserHelper::storeSegmentCabinCode::operator()`, `AIRINV::ScheduleParserHelper::storeSegmentOffPoint::operator()`, `AIRINV::ScheduleParserHelper::storeSegmentBoardingPoint::operator()`, `AIRINV::ScheduleParserHelper::storeSegmentSpecificity::operator()`, `AIRINV::ScheduleParserHelper::storeCapacity::operator()`, `AIRINV::ScheduleParserHelper::storeLegCabinCode::operator()`, `AIRINV::ScheduleParserHelper::storeElapsedTime::operator()`, `AIRINV::ScheduleParserHelper::storeOffTime::operator()`, and `AIRINV::ScheduleParserHelper::storeBoardingTime::operator()`.

AIRINV::ScheduleParserHelper::storeOperatingFlightNumber::operator>(), AIRINV::ScheduleParserHelper::storeOperatingAirlineCode::operator>(), AIRINV::ScheduleParserHelper::storeLegOffPoint::operator>(), AIRINV::ScheduleParserHelper::storeLegBoardingPoint::operator>(), AIRINV::ScheduleParserHelper::storeDow::operator>(), AIRINV::ScheduleParserHelper::storeDateRangeEnd::operator>(), AIRINV::ScheduleParserHelper::storeDateRangeStart::operator>(), AIRINV::ScheduleParserHelper::storeFlightNumber::operator(), and AIRINV::ScheduleParserHelper::storeAirlineCode::operator()).

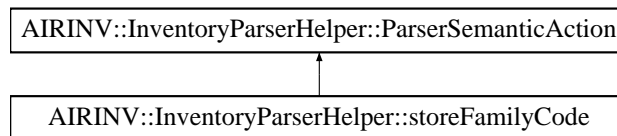
The documentation for this struct was generated from the following files:

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

## 10.145 AIRINV::InventoryParserHelper::storeFamilyCode Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeFamilyCode::



### Public Member Functions

- [storeFamilyCode](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) (int iCode) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.145.1 Detailed Description

Store the parsed family code.

Definition at line 425 of file [InventoryParserHelper.hpp](#).

#### 10.145.2 Constructor & Destructor Documentation

##### 10.145.2.1 AIRINV::InventoryParserHelper::storeFamilyCode::storeFamilyCode ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 737 of file [InventoryParserHelper.cpp](#).

### 10.145.3 Member Function Documentation

#### 10.145.3.1 void AIRINV::InventoryParserHelper::storeFamilyCode::operator() (int *iCode*) const

Actor Function (functor).

Definition at line 742 of file InventoryParserHelper.cpp.

References AIRINV::FareFamilyStruct::\_familyCode, AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::SegmentCabinStruct::\_itFareFamily, and AIRINV::FlightDateStruct::\_itSegmentCabin.

### 10.145.4 Member Data Documentation

#### 10.145.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParser-

Helper::storeSnapshotDate::operator()().

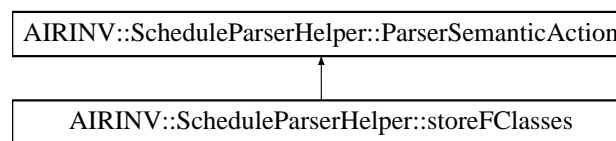
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.146 AIRINV::ScheduleParserHelper::storeFClasses Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeFClasses::



### Public Member Functions

- [storeFClasses](#) ([FlightPeriodStruct](#) &)
- [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.146.1 Detailed Description

Store the parsed list of class codes (for families).

Definition at line 216 of file ScheduleParserHelper.hpp.

#### 10.146.2 Constructor & Destructor Documentation

##### 10.146.2.1 AIRINV::ScheduleParserHelper::storeFClasses::storeFClasses ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 409 of file ScheduleParserHelper.cpp.

#### 10.146.3 Member Function Documentation

##### 10.146.3.1 void AIRINV::ScheduleParserHelper::storeFClasses::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 414 of file ScheduleParserHelper.cpp.

References AIRINV::FlightPeriodStruct::\_areSegmentDefinitionsSpecific, AIRINV::FareFamilyStruct::\_familyCode, AIRINV::FareFamilyStruct::\_ffDisutilityCurveKey, AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, AIRINV::FareFamilyStruct::\_frat5CurveKey, AIRINV::SegmentCabinStruct::\_itFareFamily, AIRINV::FlightPeriodStruct::\_itSegment, AIRINV::FlightPeriodStruct::\_itSegmentCabin, and AIRINV::FlightPeriodStruct::addFareFamily().

#### 10.146.4 Member Data Documentation

##### 10.146.4.1 FlightPeriodStruct & AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flight-Period [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

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

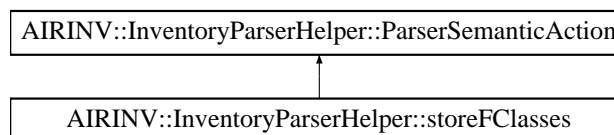
The documentation for this struct was generated from the following files:

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

## 10.147 AIRINV::InventoryParserHelper::storeFClasses Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeFClasses:



#### Public Member Functions

- [storeFClasses](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

## Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

### 10.147.1 Detailed Description

Store the parsed list of class codes (for families).

Definition at line 433 of file InventoryParserHelper.hpp.

### 10.147.2 Constructor & Destructor Documentation

#### 10.147.2.1 AIRINV::InventoryParserHelper::storeFClasses::storeFClasses ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 749 of file InventoryParserHelper.cpp.

### 10.147.3 Member Function Documentation

#### 10.147.3.1 void AIRINV::InventoryParserHelper::storeFClasses::operator() ([iterator\\_t](#) *iStr*, [iterator\\_t](#) *iStrEnd*) const

Actor Function (functor).

Definition at line 754 of file InventoryParserHelper.cpp.

References [AIRINV::SegmentStruct::\\_cabinList](#), [AIRINV::BookingClassStruct::\\_classCode](#), [AIRINV::FareFamilyStruct::\\_classes](#), [AIRINV::FareFamilyStruct::\\_classList](#), [AIRINV::SegmentCabinStruct::\\_fareFamilies](#), [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), [AIRINV::FlightDateStruct::\\_itBookingClass](#), [AIRINV::SegmentCabinStruct::\\_itFareFamily](#), [AIRINV::FlightDateStruct::\\_itSegment](#), and [AIRINV::FlightDateStruct::\\_itSegmentCabin](#).

### 10.147.4 Member Data Documentation

#### 10.147.4.1 [FlightDateStruct](#)& [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#) [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by [AIRINV::InventoryParserHelper::doEndFlightDate::operator\(\)](#), [operator\(\)](#), [AIRINV::InventoryParserHelper::storeFamilyCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeRevenueAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeSegmentAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassETB::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeOverbooking::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNoShow::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNego::operator\(\)](#), [AIRINV::InventoryParserHelper::storeProtection::operator\(\)](#), [AIRINV::InventoryParserHelper::storeCumulatedProtection::operator\(\)](#), [AIRINV::InventoryParserHelper::storeParentSubclassCode::operator\(\)](#), and [AIRINV::InventoryParserHelper::storeParentClassCode::operator\(\)](#).

AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

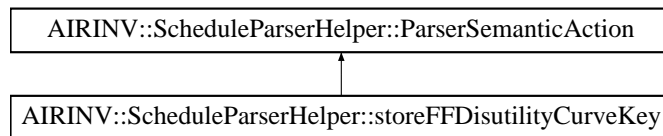
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.148 AIRINV::ScheduleParserHelper::storeFFDisutilityCurveKey Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeFFDisutilityCurveKey::



### Public Member Functions

- [storeFFDisutilityCurveKey](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

### 10.148.1 Detailed Description

Store the FFDisutility curve key.

Definition at line 208 of file ScheduleParserHelper.hpp.

### 10.148.2 Constructor & Destructor Documentation

#### 10.148.2.1 AIRINV::ScheduleParserHelper::storeFFDisutilityCurveKey::storeFFDisutilityCurveKey (FlightPeriodStruct &)

Actor Constructor.

Definition at line 396 of file ScheduleParserHelper.cpp.

### 10.148.3 Member Function Documentation

#### 10.148.3.1 void AIRINV::ScheduleParserHelper::storeFFDisutilityCurveKey::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 401 of file ScheduleParserHelper.cpp.

References AIRINV::FareFamilyStruct::\_ffDisutilityCurveKey, AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, AIRINV::SegmentCabinStruct::\_itFareFamily, and AIRINV::FlightPeriodStruct::\_itSegmentCabin.

### 10.148.4 Member Data Documentation

#### 10.148.4.1 FlightPeriodStruct& AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

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

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

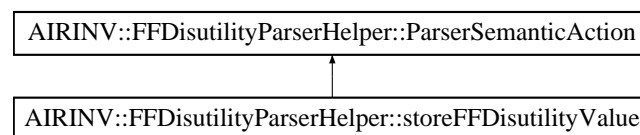


- [airinv/command/ScheduleParserHelper.hpp](#)
- [airinv/command/ScheduleParserHelper.cpp](#)

## 10.149 AIRINV::FFDisutilityParserHelper::storeFFDisutilityValue Struct Reference

```
#include <airinv/command/FFDisutilityParserHelper.hpp>
```

Inheritance diagram for AIRINV::FFDisutilityParserHelper::storeFFDisutilityValue::



### Public Member Functions

- [storeFFDisutilityValue](#) (FFDisutilityStruct &)
- void [operator\(\)](#) (double iReal) const

### Public Attributes

- [FFDisutilityStruct](#) & [\\_ffDisutility](#)

### 10.149.1 Detailed Description

Store the FFDisutility value.

Definition at line 53 of file FFDisutilityParserHelper.hpp.

### 10.149.2 Constructor & Destructor Documentation

#### 10.149.2.1 AIRINV::FFDisutilityParserHelper::storeFFDisutilityValue::storeFFDisutilityValue (FFDisutilityStruct &)

Actor Constructor.

Definition at line 56 of file FFDisutilityParserHelper.cpp.

### 10.149.3 Member Function Documentation

#### 10.149.3.1 void AIRINV::FFDisutilityParserHelper::storeFFDisutilityValue::operator() (double *i-Real*) const

Actor Function (functor).

Definition at line 61 of file FFDisutilityParserHelper.cpp.

References [AIRINV::FFDisutilityStruct::\\_curve](#), [AIRINV::FFDisutilityStruct::\\_dtd](#), and [AIRINV::FFDisutilityParserHelper::ParserSemanticAction::\\_ffDisutility](#).

#### 10.149.4 Member Data Documentation

##### 10.149.4.1 FFDisutilityStruct& AIRINV::FFDisutilityParserHelper::ParserSemanticAction::\_ffDisutility [inherited]

Actor Context.

Definition at line 33 of file FFDisutilityParserHelper.hpp.

Referenced by AIRINV::FFDisutilityParserHelper::doEndCurve::operator>(), operator(), AIRINV::FFDisutilityParserHelper::storeDTD::operator(), and AIRINV::FFDisutilityParserHelper::storeCurveKey::operator().

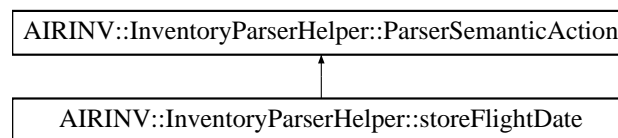
The documentation for this struct was generated from the following files:

- [airinv/command/FFDisutilityParserHelper.hpp](#)
- [airinv/command/FFDisutilityParserHelper.cpp](#)

### 10.150 AIRINV::InventoryParserHelper::storeFlightDate Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeFlightDate::



#### Public Member Functions

- [storeFlightDate](#) ([FlightDateStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

#### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.150.1 Detailed Description

Store the flight date.

Definition at line 61 of file InventoryParserHelper.hpp.

#### 10.150.2 Constructor & Destructor Documentation

##### 10.150.2.1 AIRINV::InventoryParserHelper::storeFlightDate::storeFlightDate ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 80 of file InventoryParserHelper.cpp.

### 10.150.3 Member Function Documentation

#### 10.150.3.1 void AIRINV::InventoryParserHelper::storeFlightDate::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 85 of file InventoryParserHelper.cpp.

References AIRINV::FlightDateStruct::\_flightDate, AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, and AIRINV::FlightDateStruct::getDate().

### 10.150.4 Member Data Documentation

#### 10.150.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

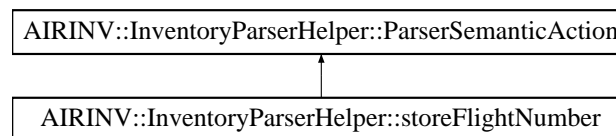
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.151 AIRINV::InventoryParserHelper::storeFlightNumber Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeFlightNumber::



### Public Member Functions

- [storeFlightNumber](#) ([FlightDateStruct](#) &)
- void [operator\(\)](#) (unsigned int iNumber) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.151.1 Detailed Description

Store the parsed flight number.

Definition at line 53 of file InventoryParserHelper.hpp.

#### 10.151.2 Constructor & Destructor Documentation

##### 10.151.2.1 AIRINV::InventoryParserHelper::storeFlightNumber::storeFlightNumber ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 70 of file InventoryParserHelper.cpp.

#### 10.151.3 Member Function Documentation

##### 10.151.3.1 void AIRINV::InventoryParserHelper::storeFlightNumber::operator() (unsigned int *i-Number*) const

Actor Function (functor).

Definition at line 75 of file InventoryParserHelper.cpp.

References [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), and [AIRINV::FlightDateStruct::\\_flightNumber](#).

#### 10.151.4 Member Data Documentation

##### 10.151.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flight-Date [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvaibility::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

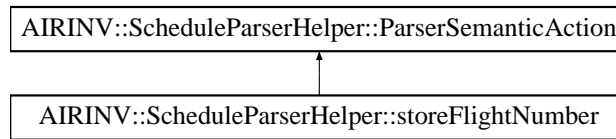
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.152 AIRINV::ScheduleParserHelper::storeFlightNumber Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeFlightNumber::



## Public Member Functions

- [storeFlightNumber](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) (unsigned int iNumber) const

## Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

### 10.152.1 Detailed Description

Store the parsed flight number.

Definition at line 45 of file `ScheduleParserHelper.hpp`.

### 10.152.2 Constructor & Destructor Documentation

#### 10.152.2.1 AIRINV::ScheduleParserHelper::storeFlightNumber::storeFlightNumber ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 51 of file `ScheduleParserHelper.cpp`.

### 10.152.3 Member Function Documentation

#### 10.152.3.1 void AIRINV::ScheduleParserHelper::storeFlightNumber::operator() (unsigned int *iNumber*) const

Actor Function (functor).

Definition at line 56 of file `ScheduleParserHelper.cpp`.

References `AIRINV::FlightPeriodStruct::_flightNumber`, and `AIRINV::ScheduleParserHelper::ParserSemanticAction::_flightPeriod`.

### 10.152.4 Member Data Documentation

#### 10.152.4.1 [FlightPeriodStruct&](#) [AIRINV::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#) [inherited]

Actor Context.

Definition at line 33 of file `ScheduleParserHelper.hpp`.

Referenced by `AIRINV::ScheduleParserHelper::doEndFlight::operator()`, `AIRINV::ScheduleParserHelper::storeFClasses::operator()`, `AIRINV::ScheduleParserHelper::storeFFDisutilityCurveKey::operator()`, and `AIRINV::ScheduleParserHelper::storeFRAT5CurveKey::operator()`.

AIRINV::ScheduleParserHelper::storeFamilyCode::operator(), AIRINV::ScheduleParserHelper::storeClasses::operator(), AIRINV::ScheduleParserHelper::storeSegmentCabinCode::operator(), AIRINV::ScheduleParserHelper::storeSegmentOffPoint::operator(), AIRINV::ScheduleParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::ScheduleParserHelper::storeSegmentSpecificity::operator(), AIRINV::ScheduleParserHelper::storeCapacity::operator(), AIRINV::ScheduleParserHelper::storeLegCabinCode::operator(), AIRINV::ScheduleParserHelper::storeElapsedTime::operator(), AIRINV::ScheduleParserHelper::storeOffTime::operator(), AIRINV::ScheduleParserHelper::storeBoardingTime::operator(), AIRINV::ScheduleParserHelper::storeOperatingFlightNumber::operator(), AIRINV::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRINV::ScheduleParserHelper::storeLegOffPoint::operator(), AIRINV::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRINV::ScheduleParserHelper::storeDow::operator(), AIRINV::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRINV::ScheduleParserHelper::storeDateRangeStart::operator(), operator(), and AIRINV::ScheduleParserHelper::storeAirlineCode::operator().

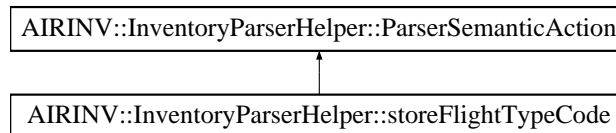
The documentation for this struct was generated from the following files:

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

## 10.153 AIRINV::InventoryParserHelper::storeFlightTypeCode Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeFlightTypeCode:



### Public Member Functions

- [storeFlightTypeCode](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.153.1 Detailed Description

Store the flight type code.

Definition at line 69 of file [InventoryParserHelper.hpp](#).

#### 10.153.2 Constructor & Destructor Documentation

##### 10.153.2.1 AIRINV::InventoryParserHelper::storeFlightTypeCode::storeFlightTypeCode ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 91 of file InventoryParserHelper.cpp.

### 10.153.3 Member Function Documentation

#### 10.153.3.1 void AIRINV::InventoryParserHelper::storeFlightTypeCode::operator() (iterator\_t i-Str, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 96 of file InventoryParserHelper.cpp.

References AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_flightTypeCode, and AIRINV::FlightTypeCode::getCode().

### 10.153.4 Member Data Documentation

#### 10.153.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::store-



LegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

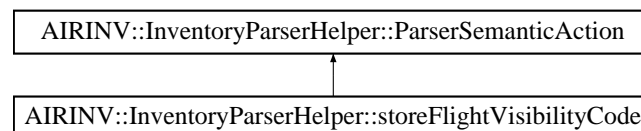
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.154 AIRINV::InventoryParserHelper::storeFlightVisibilityCode Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeFlightVisibilityCode::



### Public Member Functions

- [storeFlightVisibilityCode](#) ([FlightDateStruct](#) &)
- [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.154.1 Detailed Description

Store the flight visibility code.

Definition at line 77 of file InventoryParserHelper.hpp.

#### 10.154.2 Constructor & Destructor Documentation

##### 10.154.2.1 AIRINV::InventoryParserHelper::storeFlightVisibilityCode::storeFlightVisibilityCode ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 106 of file InventoryParserHelper.cpp.

#### 10.154.3 Member Function Documentation

##### 10.154.3.1 void AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 111 of file InventoryParserHelper.cpp.

References AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_flightVisibilityCode, and AIRINV::FlightVisibilityCode::getCode().

#### 10.154.4 Member Data Documentation

##### 10.154.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

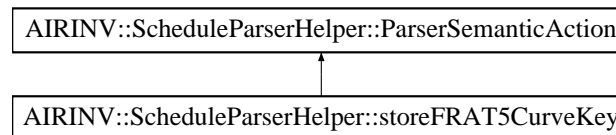
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.155 AIRINV::ScheduleParserHelper::storeFRAT5CurveKey Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeFRAT5CurveKey::



### Public Member Functions

- [storeFRAT5CurveKey](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.155.1 Detailed Description

Store the FRAT5 curve key.

Definition at line 200 of file ScheduleParserHelper.hpp.

#### 10.155.2 Constructor & Destructor Documentation

##### 10.155.2.1 AIRINV::ScheduleParserHelper::storeFRAT5CurveKey::storeFRAT5CurveKey ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 382 of file ScheduleParserHelper.cpp.

#### 10.155.3 Member Function Documentation

##### 10.155.3.1 void AIRINV::ScheduleParserHelper::storeFRAT5CurveKey::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 387 of file ScheduleParserHelper.cpp.

References [AIRINV::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#), [AIRINV::FareFamilyStruct::\\_frat5CurveKey](#), [AIRINV::SegmentCabinStruct::\\_itFareFamily](#), and [AIRINV::FlightPeriodStruct::\\_itSegmentCabin](#).

#### 10.155.4 Member Data Documentation

##### 10.155.4.1 [FlightPeriodStruct](#)& [AIRINV::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#) [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

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

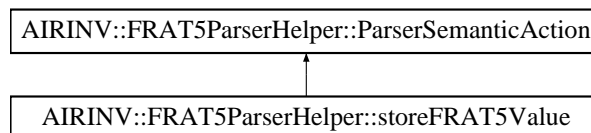
The documentation for this struct was generated from the following files:

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

## 10.156 AIRINV::FRAT5ParserHelper::storeFRAT5Value Struct Reference

```
#include <airinv/command/FRAT5ParserHelper.hpp>
```

Inheritance diagram for AIRINV::FRAT5ParserHelper::storeFRAT5Value::



### Public Member Functions

- [storeFRAT5Value](#) ([FRAT5Struct](#) &)
- void [operator\(\)](#) (double iReal) const

### Public Attributes

- [FRAT5Struct](#) & [\\_frat5](#)

#### 10.156.1 Detailed Description

Store the FRAT5 value.

Definition at line 53 of file FRAT5ParserHelper.hpp.

### 10.156.2 Constructor & Destructor Documentation

#### 10.156.2.1 AIRINV::FRAT5ParserHelper::storeFRAT5Value::storeFRAT5Value ([FRAT5Struct &](#))

Actor Constructor.

Definition at line 57 of file FRAT5ParserHelper.cpp.

### 10.156.3 Member Function Documentation

#### 10.156.3.1 void AIRINV::FRAT5ParserHelper::storeFRAT5Value::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 62 of file FRAT5ParserHelper.cpp.

References [AIRINV::FRAT5Struct::\\_curve](#), [AIRINV::FRAT5Struct::\\_dtd](#), and [AIRINV::FRAT5ParserHelper::ParserSemanticAction::\\_frat5](#).

### 10.156.4 Member Data Documentation

#### 10.156.4.1 [FRAT5Struct&](#) [AIRINV::FRAT5ParserHelper::ParserSemanticAction::\\_frat5](#) [inherited]

Actor Context.

Definition at line 33 of file FRAT5ParserHelper.hpp.

Referenced by [AIRINV::FRAT5ParserHelper::doEndCurve::operator\(\)](#), [operator\(\)](#), [AIRINV::FRAT5ParserHelper::storeDTD::operator\(\)](#), and [AIRINV::FRAT5ParserHelper::storeCurveKey::operator\(\)](#).

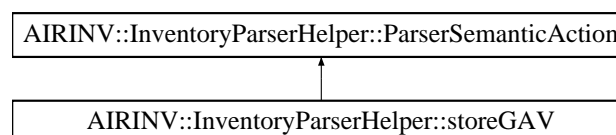
The documentation for this struct was generated from the following files:

- [airinv/command/FRAT5ParserHelper.hpp](#)
- [airinv/command/FRAT5ParserHelper.cpp](#)

## 10.157 AIRINV::InventoryParserHelper::storeGAV Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeGAV::



### Public Member Functions

- [storeGAV](#) ([FlightDateStruct &](#))
- void [operator\(\)](#) (double iReal) const

**Public Attributes**

- [FlightDateStruct](#) & [\\_flightDate](#)

**10.157.1 Detailed Description**

Store the parsed Gross Availability (GAV).

Definition at line 197 of file InventoryParserHelper.hpp.

**10.157.2 Constructor & Destructor Documentation****10.157.2.1 AIRINV::InventoryParserHelper::storeGAV::storeGAV ([FlightDateStruct](#) &)**

Actor Constructor.

Definition at line 339 of file InventoryParserHelper.cpp.

**10.157.3 Member Function Documentation****10.157.3.1 void AIRINV::InventoryParserHelper::storeGAV::operator() (double *iReal*) const**

Actor Function (functor).

Definition at line 344 of file InventoryParserHelper.cpp.

References [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), [AIRINV::LegCabinStruct::\\_gav](#), and [AIRINV::FlightDateStruct::\\_itLegCabin](#).

**10.157.4 Member Data Documentation****10.157.4.1 [FlightDateStruct&](#) [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#) [inherited]**

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by [AIRINV::InventoryParserHelper::doEndFlightDate::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFCClasses::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFamilyCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeRevenueAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeSegmentAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassETB::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeOverbooking::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNoShow::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNego::operator\(\)](#), [AIRINV::InventoryParserHelper::storeProtection::operator\(\)](#), [AIRINV::InventoryParserHelper::storeCumulatedProtection::operator\(\)](#), [AIRINV::InventoryParserHelper::storeParentSubclassCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeParentClassCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeSubclassCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator\(\)](#), [AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator\(\)](#), and [AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator\(\)](#).

AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

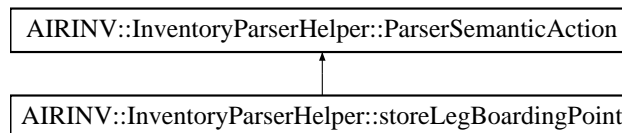
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.158 AIRINV::InventoryParserHelper::storeLegBoardingPoint Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeLegBoardingPoint::



### Public Member Functions

- [storeLegBoardingPoint](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.158.1 Detailed Description

Store the parsed leg boarding point.

Definition at line 85 of file [InventoryParserHelper.hpp](#).

## 10.158.2 Constructor & Destructor Documentation

### 10.158.2.1 AIRINV::InventoryParserHelper::storeLegBoardingPoint::storeLegBoardingPoint (FlightDateStruct &)

Actor Constructor.

Definition at line 121 of file InventoryParserHelper.cpp.

## 10.158.3 Member Function Documentation

### 10.158.3.1 void AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 126 of file InventoryParserHelper.cpp.

References AIRINV::FlightDateStruct::\_airlineCode, AIRINV::LegStruct::\_airlineCode, AIRINV::LegStruct::\_boardingPoint, AIRINV::LegCabinStruct::\_bucketList, AIRINV::LegCabinStruct::\_cabinCode, AIRINV::LegStruct::\_cabinList, AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_flightNumber, AIRINV::LegStruct::\_flightNumber, AIRINV::FlightDateStruct::\_itBucket, AIRINV::FlightDateStruct::\_itLeg, AIRINV::FlightDateStruct::\_itLegCabin, AIRINV::FlightDateStruct::\_legList, AIRINV::BucketStruct::\_yieldRangeUpperValue, and AIRINV::FlightDateStruct::addAirport().

## 10.158.4 Member Data Documentation

### 10.158.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::store-



ACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

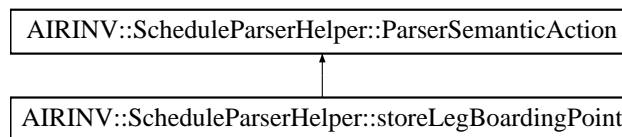
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.159 AIRINV::ScheduleParserHelper::storeLegBoardingPoint Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeLegBoardingPoint::



### Public Member Functions

- [storeLegBoardingPoint](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.159.1 Detailed Description

Store the parsed leg boarding point.

Definition at line 77 of file [ScheduleParserHelper.hpp](#).

## 10.159.2 Constructor & Destructor Documentation

### 10.159.2.1 AIRINV::ScheduleParserHelper::storeLegBoardingPoint::storeLegBoardingPoint (FlightPeriodStruct &)

Actor Constructor.

Definition at line 117 of file ScheduleParserHelper.cpp.

## 10.159.3 Member Function Documentation

### 10.159.3.1 void AIRINV::ScheduleParserHelper::storeLegBoardingPoint::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 122 of file ScheduleParserHelper.cpp.

References AIRINV::LegStruct::\_boardingPoint, AIRINV::LegStruct::\_cabinList, AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, AIRINV::FlightPeriodStruct::\_itLeg, AIRINV::FlightPeriodStruct::\_legAlreadyDefined, AIRINV::FlightPeriodStruct::\_legList, and AIRINV::FlightPeriodStruct::addAirport().

## 10.159.4 Member Data Documentation

### 10.159.4.1 FlightPeriodStruct& AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

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

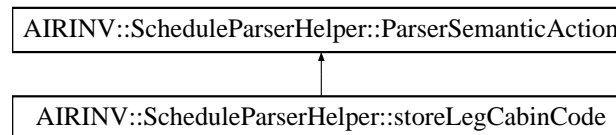
The documentation for this struct was generated from the following files:

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

## 10.160 AIRINV::ScheduleParserHelper::storeLegCabinCode Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeLegCabinCode::



### Public Member Functions

- [storeLegCabinCode](#) ([FlightPeriodStruct](#) &)
- void [operator\(\)](#) (char iChar) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.160.1 Detailed Description

Store the parsed leg cabin code.

Definition at line 133 of file ScheduleParserHelper.hpp.

#### 10.160.2 Constructor & Destructor Documentation

##### 10.160.2.1 AIRINV::ScheduleParserHelper::storeLegCabinCode::storeLegCabinCode ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 250 of file ScheduleParserHelper.cpp.

#### 10.160.3 Member Function Documentation

##### 10.160.3.1 void AIRINV::ScheduleParserHelper::storeLegCabinCode::operator() (char iChar) const

Actor Function (functor).

Definition at line 255 of file ScheduleParserHelper.cpp.

References [AIRINV::LegCabinStruct::\\_cabinCode](#), [AIRINV::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#), and [AIRINV::FlightPeriodStruct::\\_itLegCabin](#).

#### 10.160.4 Member Data Documentation

##### 10.160.4.1 [FlightPeriodStruct](#)& AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

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

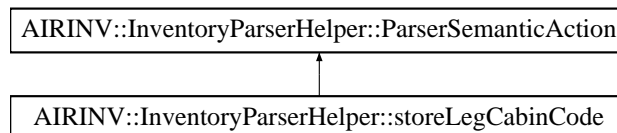
The documentation for this struct was generated from the following files:

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

## 10.161 AIRINV::InventoryParserHelper::storeLegCabinCode Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeLegCabinCode::



### Public Member Functions

- [storeLegCabinCode](#) ([FlightDateStruct](#) &)
- void [operator\(\)](#) (char iChar) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.161.1 Detailed Description

Store the parsed leg cabin code.

Definition at line 149 of file InventoryParserHelper.hpp.

## 10.161.2 Constructor & Destructor Documentation

### 10.161.2.1 AIRINV::InventoryParserHelper::storeLegCabinCode::storeLegCabinCode ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 255 of file InventoryParserHelper.cpp.

## 10.161.3 Member Function Documentation

### 10.161.3.1 void AIRINV::InventoryParserHelper::storeLegCabinCode::operator() (char *iChar*) const

Actor Function (functor).

Definition at line 260 of file InventoryParserHelper.cpp.

References AIRINV::LegCabinStruct::\_bucketList, AIRINV::LegCabinStruct::\_cabinCode, AIRINV::LegStruct::\_cabinList, AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_itBucket, AIRINV::FlightDateStruct::\_itLeg, AIRINV::FlightDateStruct::\_itLegCabin, and AIRINV::BucketStruct::\_yieldRangeUpperValue.

## 10.161.4 Member Data Documentation

### 10.161.4.1 [FlightDateStruct&](#) AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFCClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::store-

AU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

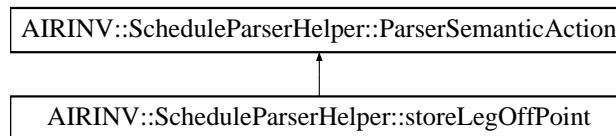
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.162 AIRINV::ScheduleParserHelper::storeLegOffPoint Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeLegOffPoint::



### Public Member Functions

- [storeLegOffPoint](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.162.1 Detailed Description

Store the parsed leg off point.

Definition at line 85 of file [ScheduleParserHelper.hpp](#).

#### 10.162.2 Constructor & Destructor Documentation

##### 10.162.2.1 AIRINV::ScheduleParserHelper::storeLegOffPoint::storeLegOffPoint ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 146 of file [ScheduleParserHelper.cpp](#).

### 10.162.3 Member Function Documentation

#### 10.162.3.1 void AIRINV::ScheduleParserHelper::storeLegOffPoint::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 151 of file ScheduleParserHelper.cpp.

References AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, AIRINV::FlightPeriodStruct::\_itLeg, AIRINV::LegStruct::\_offPoint, and AIRINV::FlightPeriodStruct::addAirport().

### 10.162.4 Member Data Documentation

#### 10.162.4.1 FlightPeriodStruct& AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

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

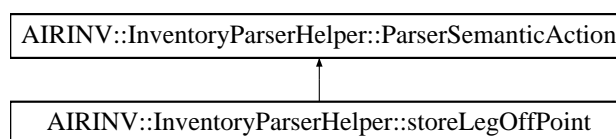
The documentation for this struct was generated from the following files:

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

## 10.163 AIRINV::InventoryParserHelper::storeLegOffPoint Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeLegOffPoint::



## Public Member Functions

- [storeLegOffPoint](#) ([FlightDateStruct](#) &)
- void [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

## Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

### 10.163.1 Detailed Description

Store the parsed leg off point.

Definition at line 93 of file InventoryParserHelper.hpp.

### 10.163.2 Constructor & Destructor Documentation

#### 10.163.2.1 AIRINV::InventoryParserHelper::storeLegOffPoint::storeLegOffPoint ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 160 of file InventoryParserHelper.cpp.

### 10.163.3 Member Function Documentation

#### 10.163.3.1 void AIRINV::InventoryParserHelper::storeLegOffPoint::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 165 of file InventoryParserHelper.cpp.

References [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), [AIRINV::FlightDateStruct::\\_itLeg](#), [AIRINV::LegStruct::\\_offPoint](#), and [AIRINV::FlightDateStruct::addAirport\(\)](#).

### 10.163.4 Member Data Documentation

#### 10.163.4.1 [FlightDateStruct&](#) [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#) [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by [AIRINV::InventoryParserHelper::doEndFlightDate::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFClasses::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFamilyCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeRevenueAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeSegmentAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassETB::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator\(\)](#), and [AIRINV::InventoryParserHelper::storeNbOfBkgs::operator\(\)](#).



AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

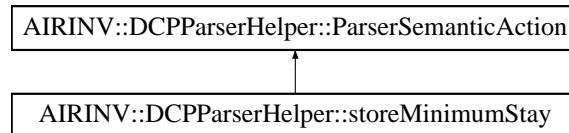
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.164 AIRINV::DCPParserHelper::storeMinimumStay Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::storeMinimumStay::



### Public Member Functions

- [storeMinimumStay](#) (DCPRuleStruct &)
- void [operator\(\)](#) (unsigned int, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

### Public Attributes

- DCPRuleStruct & [\\_DCPRule](#)

### 10.164.1 Detailed Description

Store the parsed minimum stay.

Definition at line 178 of file DCPParserHelper.hpp.

### 10.164.2 Constructor & Destructor Documentation

#### 10.164.2.1 AIRINV::DCPParserHelper::storeMinimumStay::storeMinimumStay (DCPRuleStruct &)

Actor Constructor.

Definition at line 299 of file DCPParserHelper.cpp.

### 10.164.3 Member Function Documentation

#### 10.164.3.1 void AIRINV::DCPParserHelper::storeMinimumStay::operator() (unsigned int, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

Actor Function (functor).

Definition at line 304 of file DCPParserHelper.cpp.

References AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule.

### 10.164.4 Member Data Documentation

#### 10.164.4.1 DCPRuleStruct& AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule [inherited]

Actor Context.

Definition at line 34 of file DCPParserHelper.hpp.

Referenced by AIRINV::DCPParserHelper::doEndDCP::operator(), AIRINV::DCPParserHelper::storeClass::operator(), AIRINV::DCPParserHelper::storeAirlineCode::operator(), AIRINV::DCPParserHelper::storeDCP::operator(), operator(), AIRINV::DCPParserHelper::storeNonRefundable::operator(), AIRINV::DCPParserHelper::storeChangeFees::operator(), AIRINV::DCPParserHelper::storeSaturdayStay::operator(), AIRINV::DCPParserHelper::storeAdvancePurchase::operator(), AIRINV::DCPParserHelper::storeChannel::operator(), AIRINV::DCPParserHelper::storeCabinCode::operator(), AIRINV::DCPParserHelper::storePOS::operator(), AIRINV::DCPParserHelper::storeEndRangeTime::operator(), AIRINV::DCPParserHelper::storeStartRangeTime::operator(), AIRINV::DCPParserHelper::storeDateRangeEnd::operator(), AIRINV::DCPParserHelper::storeDateRangeStart::operator(), AIRINV::DCPParserHelper::storeDestination::operator(), AIRINV::DCPParserHelper::storeOrigin::operator(), and AIRINV::DCPParserHelper::storeDCPID::operator().

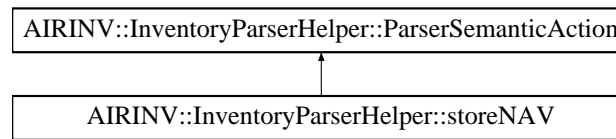
The documentation for this struct was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.165 AIRINV::InventoryParserHelper::storeNAV Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeNAV::



### Public Member Functions

- [storeNAV](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) (double iReal) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.165.1 Detailed Description

Store the parsed Net Availability (NAV).

Definition at line 189 of file InventoryParserHelper.hpp.

#### 10.165.2 Constructor & Destructor Documentation

##### 10.165.2.1 AIRINV::InventoryParserHelper::storeNAV::storeNAV ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 328 of file InventoryParserHelper.cpp.

#### 10.165.3 Member Function Documentation

##### 10.165.3.1 void AIRINV::InventoryParserHelper::storeNAV::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 333 of file InventoryParserHelper.cpp.

References [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), [AIRINV::FlightDateStruct::\\_itLegCabin](#), and [AIRINV::LegCabinStruct::\\_nav](#).

#### 10.165.4 Member Data Documentation

##### 10.165.4.1 [FlightDateStruct](#)& [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#) [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by [AIRINV::InventoryParserHelper::doEndFlightDate::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFClasses::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFamilyCode::operator\(\)](#), and [AIRINV::InventoryParserHelper::storeRevenueAvailability::operator\(\)](#).

AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNegotiation::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

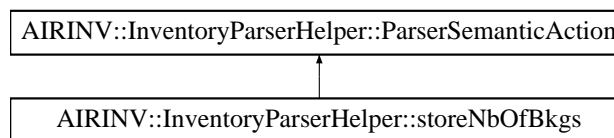
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.166 AIRINV::InventoryParserHelper::storeNbOfBkgs Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeNbOfBkgs::



## Public Member Functions

- [storeNbOfBkgs](#) ([FlightDateStruct](#) &)
- void [operator\(\)](#) (double iReal) const

## Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

### 10.166.1 Detailed Description

Store the parsed number of bookings (at booking class level).

Definition at line 349 of file InventoryParserHelper.hpp.

### 10.166.2 Constructor & Destructor Documentation

#### 10.166.2.1 AIRINV::InventoryParserHelper::storeNbOfBkgs::storeNbOfBkgs ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 634 of file InventoryParserHelper.cpp.

### 10.166.3 Member Function Documentation

#### 10.166.3.1 void AIRINV::InventoryParserHelper::storeNbOfBkgs::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 639 of file InventoryParserHelper.cpp.

References [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), [AIRINV::FlightDateStruct::\\_itBookingClass](#), and [AIRINV::BookingClassStruct::\\_nbOfBookings](#).

### 10.166.4 Member Data Documentation

#### 10.166.4.1 [FlightDateStruct&](#) [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#) [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by [AIRINV::InventoryParserHelper::doEndFlightDate::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFClasses::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFamilyCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeRevenueAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeSegmentAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassETB::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator\(\)](#), [operator\(\)](#), [AIRINV::InventoryParserHelper::storeOverbooking::operator\(\)](#),

AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

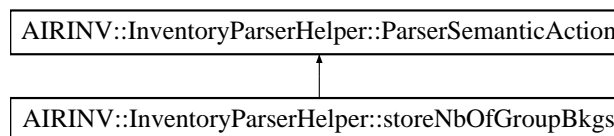
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.167 AIRINV::InventoryParserHelper::storeNbOfGroupBkgs Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::



### Public Member Functions

- [storeNbOfGroupBkgs](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) (double iReal) const

**Public Attributes**

- [FlightDateStruct](#) & [\\_flightDate](#)

**10.167.1 Detailed Description**

Store the parsed number of group bookings (at booking class level).

Definition at line 357 of file InventoryParserHelper.hpp.

**10.167.2 Constructor & Destructor Documentation****10.167.2.1 AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::storeNbOfGroupBkgs([FlightDateStruct](#) &)**

Actor Constructor.

Definition at line 645 of file InventoryParserHelper.cpp.

**10.167.3 Member Function Documentation****10.167.3.1 void AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator() (double *i-Real*) const**

Actor Function (functor).

Definition at line 650 of file InventoryParserHelper.cpp.

References [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), [AIRINV::FlightDateStruct::\\_itBookingClass](#), and [AIRINV::BookingClassStruct::\\_nbOfGroupBookings](#).

**10.167.4 Member Data Documentation****10.167.4.1 [FlightDateStruct](#)& [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#) [inherited]**

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by [AIRINV::InventoryParserHelper::doEndFlightDate::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFCClasses::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFamilyCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeRevenueAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeSegmentAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassETB::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator\(\)](#), [operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeOverbooking::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNoShow::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNego::operator\(\)](#), [AIRINV::InventoryParserHelper::storeProtection::operator\(\)](#), [AIRINV::InventoryParserHelper::storeCumulatedProtection::operator\(\)](#), [AIRINV::InventoryParserHelper::storeParentSubclassCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeParentClassCode::operator\(\)](#), and [AIRINV::InventoryParserHelper::storeSubclassCode::operator\(\)](#).



AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

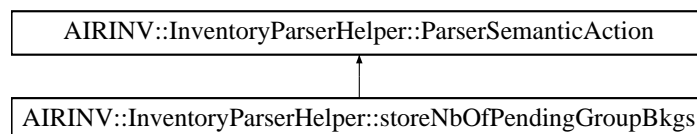
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.168 AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::



### Public Member Functions

- [storeNbOfPendingGroupBkgs](#) ([FlightDateStruct](#) &)
- [operator\(\)](#) (double iReal) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)



### 10.168.1 Detailed Description

Store the parsed number of pending group bookings (at booking class level).

Definition at line 365 of file InventoryParserHelper.hpp.

### 10.168.2 Constructor & Destructor Documentation

#### 10.168.2.1 AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::storeNbOfPendingGroupBkgs (FlightDateStruct &)

Actor Constructor.

Definition at line 657 of file InventoryParserHelper.cpp.

### 10.168.3 Member Function Documentation

#### 10.168.3.1 void AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator() (double iReal) const

Actor Function (functor).

Definition at line 662 of file InventoryParserHelper.cpp.

References AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_itBookingClass, and AIRINV::BookingClassStruct::\_nbOfPendingGroupBookings.

### 10.168.4 Member Data Documentation

#### 10.168.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::Inventory-

ParserHelper::storeETB::operator>(), AIRINV::InventoryParserHelper::storeACP::operator>(),  
 AIRINV::InventoryParserHelper::storeGAV::operator>(), AIRINV::InventoryParserHelper::store-  
 NAV::operator>(), AIRINV::InventoryParserHelper::storeBookingCounter::operator>(),  
 AIRINV::InventoryParserHelper::storeUPR::operator>(), AIRINV::InventoryParserHelper::store-  
 AU::operator>(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator>(),  
 AIRINV::InventoryParserHelper::storeLegCabinCode::operator>(), AIRINV::InventoryParser-  
 Helper::storeOffTime::operator>(), AIRINV::InventoryParserHelper::storeOffDate::operator>(),  
 AIRINV::InventoryParserHelper::storeBoardingTime::operator>(), AIRINV::InventoryParser-  
 Helper::storeBoardingDate::operator>(), AIRINV::InventoryParserHelper::storeOperatingFlight-  
 Number::operator>(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator>(),  
 AIRINV::InventoryParserHelper::storeLegOffPoint::operator>(), AIRINV::InventoryParserHelper::store-  
 LegBoardingPoint::operator>(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator>(),  
 AIRINV::InventoryParserHelper::storeFlightTypeCode::operator>(), AIRINV::InventoryParser-  
 Helper::storeFlightDate::operator>(), AIRINV::InventoryParserHelper::storeFlightNumber::operator>(),  
 AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParser-  
 Helper::storeSnapshotDate::operator()).

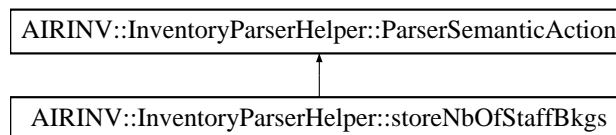
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.169 AIRINV::InventoryParserHelper::storeNbOfStaffBkgs Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::



### Public Member Functions

- [storeNbOfStaffBkgs](#) ([FlightDateStruct](#) &)
- [operator\(\)](#) (double iReal) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.169.1 Detailed Description

Store the parsed number of staff bookings (at booking class level).

Definition at line 373 of file [InventoryParserHelper.hpp](#).

## 10.169.2 Constructor & Destructor Documentation

### 10.169.2.1 AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::storeNbOfStaffBkgs ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 668 of file InventoryParserHelper.cpp.

## 10.169.3 Member Function Documentation

### 10.169.3.1 void AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 673 of file InventoryParserHelper.cpp.

References AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_itBookingClass, and AIRINV::BookingClassStruct::\_nbOfStaffBookings.

## 10.169.4 Member Data Documentation

### 10.169.4.1 [FlightDateStruct&](#) AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParser-

Helper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(),  
 AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParser-  
 Helper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlight-  
 Number::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(),  
 AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::store-  
 LegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(),  
 AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParser-  
 Helper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(),  
 AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParser-  
 Helper::storeSnapshotDate::operator().

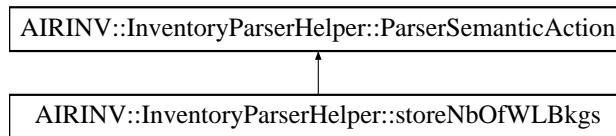
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.170 AIRINV::InventoryParserHelper::storeNbOfWLBkgs Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeNbOfWLBkgs::



### Public Member Functions

- [storeNbOfWLBkgs](#) ([FlightDateStruct](#) &)
- void [operator\(\)](#) (double iReal) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.170.1 Detailed Description

Store the parsed number of wait-list bookings (at booking class level).

Definition at line 382 of file [InventoryParserHelper.hpp](#).

#### 10.170.2 Constructor & Destructor Documentation

##### 10.170.2.1 AIRINV::InventoryParserHelper::storeNbOfWLBkgs::storeNbOfWLBkgs ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 679 of file [InventoryParserHelper.cpp](#).

### 10.170.3 Member Function Documentation

#### 10.170.3.1 void AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 684 of file InventoryParserHelper.cpp.

References AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_itBookingClass, and AIRINV::BookingClassStruct::\_nbOfWLBookings.

### 10.170.4 Member Data Documentation

#### 10.170.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeSAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParser-

Helper::storeSnapshotDate::operator()().

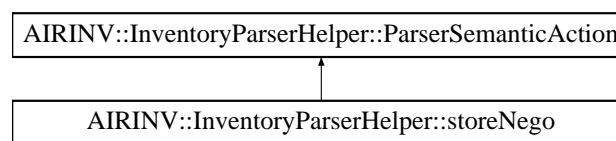
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.171 AIRINV::InventoryParserHelper::storeNego Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeNego::



### Public Member Functions

- [storeNego](#) ([FlightDateStruct](#) &)
- void [operator\(\)](#) (double *iReal*) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.171.1 Detailed Description

Store the negotiated allotment (at booking class level).

Definition at line 325 of file [InventoryParserHelper.hpp](#).

#### 10.171.2 Constructor & Destructor Documentation

##### 10.171.2.1 AIRINV::InventoryParserHelper::storeNego::storeNego ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 601 of file [InventoryParserHelper.cpp](#).

#### 10.171.3 Member Function Documentation

##### 10.171.3.1 void AIRINV::InventoryParserHelper::storeNego::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 606 of file [InventoryParserHelper.cpp](#).

References [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), [AIRINV::FlightDateStruct::\\_itBookingClass](#), and [AIRINV::BookingClassStruct::\\_nego](#).

#### 10.171.4 Member Data Documentation

##### 10.171.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flight-Date [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

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

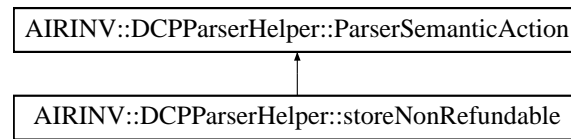
- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.172 AIRINV::DCPParserHelper::storeNonRefundable Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```



Inheritance diagram for AIRINV::DCPParserHelper::storeNonRefundable::



## Public Member Functions

- [storeNonRefundable](#) (DCPRuleStruct &)
- void [operator\(\)](#) (char, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

## Public Attributes

- DCPRuleStruct & [\\_DCPRule](#)

### 10.172.1 Detailed Description

Store the parsed refundable option

Definition at line 168 of file DCPParserHelper.hpp.

### 10.172.2 Constructor & Destructor Documentation

#### 10.172.2.1 AIRINV::DCPParserHelper::storeNonRefundable::storeNonRefundable (DCPRuleStruct &)

Actor Constructor.

Definition at line 274 of file DCPParserHelper.cpp.

### 10.172.3 Member Function Documentation

#### 10.172.3.1 void AIRINV::DCPParserHelper::storeNonRefundable::operator() (char, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

Actor Function (functor).

Definition at line 279 of file DCPParserHelper.cpp.

References AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule.

### 10.172.4 Member Data Documentation

#### 10.172.4.1 DCPRuleStruct& AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule [inherited]

Actor Context.

Definition at line 34 of file DCPParserHelper.hpp.



Referenced by AIRINV::DCPParserHelper::doEndDCP::operator(), AIRINV::DCPParserHelper::storeClass::operator(), AIRINV::DCPParserHelper::storeAirlineCode::operator(), AIRINV::DCPParserHelper::storeDCP::operator(), AIRINV::DCPParserHelper::storeMinimumStay::operator(), operator(), AIRINV::DCPParserHelper::storeChangeFees::operator(), AIRINV::DCPParserHelper::storeSaturdayStay::operator(), AIRINV::DCPParserHelper::storeAdvancePurchase::operator(), AIRINV::DCPParserHelper::storeChannel::operator(), AIRINV::DCPParserHelper::storeCabinCode::operator(), AIRINV::DCPParserHelper::storePOS::operator(), AIRINV::DCPParserHelper::storeEndRangeTime::operator(), AIRINV::DCPParserHelper::storeStartRangeTime::operator(), AIRINV::DCPParserHelper::storeDateRangeEnd::operator(), AIRINV::DCPParserHelper::storeDateRangeStart::operator(), AIRINV::DCPParserHelper::storeDestination::operator(), AIRINV::DCPParserHelper::storeOrigin::operator(), and AIRINV::DCPParserHelper::storeDCPId::operator().

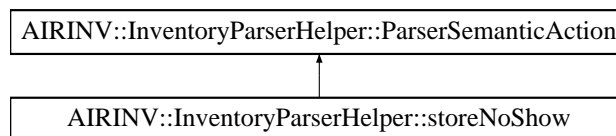
The documentation for this struct was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.173 AIRINV::InventoryParserHelper::storeNoShow Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeNoShow::



### Public Member Functions

- [storeNoShow](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) (double iReal) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.173.1 Detailed Description

Store the parsed No-Show percentage (at booking class level).

Definition at line 333 of file InventoryParserHelper.hpp.

#### 10.173.2 Constructor & Destructor Documentation

##### 10.173.2.1 AIRINV::InventoryParserHelper::storeNoShow::storeNoShow ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 612 of file InventoryParserHelper.cpp.

### 10.173.3 Member Function Documentation

#### 10.173.3.1 void AIRINV::InventoryParserHelper::storeNoShow::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 617 of file InventoryParserHelper.cpp.

References AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_itBookingClass, and AIRINV::BookingClassStruct::\_noShowPercentage.

### 10.173.4 Member Data Documentation

#### 10.173.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParser-

Helper::storeSnapshotDate::operator()().

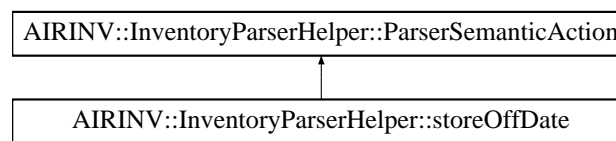
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.174 AIRINV::InventoryParserHelper::storeOffDate Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeOffDate::



### Public Member Functions

- [storeOffDate](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.174.1 Detailed Description

Store the off date.

Definition at line 133 of file [InventoryParserHelper.hpp](#).

#### 10.174.2 Constructor & Destructor Documentation

##### 10.174.2.1 AIRINV::InventoryParserHelper::storeOffDate::storeOffDate ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 232 of file [InventoryParserHelper.cpp](#).

#### 10.174.3 Member Function Documentation

##### 10.174.3.1 void AIRINV::InventoryParserHelper::storeOffDate::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 237 of file [InventoryParserHelper.cpp](#).

References [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), [AIRINV::FlightDateStruct::\\_itLeg](#), [AIRINV::LegStruct::\\_offDate](#), and [AIRINV::FlightDateStruct::getDate\(\)](#).

#### 10.174.4 Member Data Documentation

##### 10.174.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flight-Date [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvaibility::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

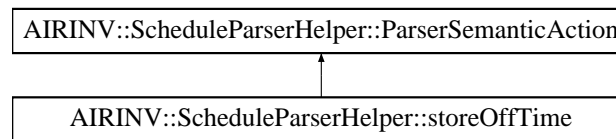
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.175 AIRINV::ScheduleParserHelper::storeOffTime Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeOffTime::



### Public Member Functions

- [storeOffTime](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.175.1 Detailed Description

Store the off time.

Definition at line 117 of file `ScheduleParserHelper.hpp`.

#### 10.175.2 Constructor & Destructor Documentation

##### 10.175.2.1 AIRINV::ScheduleParserHelper::storeOffTime::storeOffTime ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 208 of file `ScheduleParserHelper.cpp`.

#### 10.175.3 Member Function Documentation

##### 10.175.3.1 void AIRINV::ScheduleParserHelper::storeOffTime::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 213 of file `ScheduleParserHelper.cpp`.

References `AIRINV::LegStruct::_boardingDateOffset`, `AIRINV::FlightPeriodStruct::_dateOffset`, `AIRINV::ScheduleParserHelper::ParserSemanticAction::_flightPeriod`, `AIRINV::FlightPeriodStruct::_itLeg`, `AIRINV::FlightPeriodStruct::_itSeconds`, `AIRINV::LegStruct::_offTime`, and `AIRINV::FlightPeriodStruct::getTime()`.

#### 10.175.4 Member Data Documentation

##### 10.175.4.1 [FlightPeriodStruct](#)& AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

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

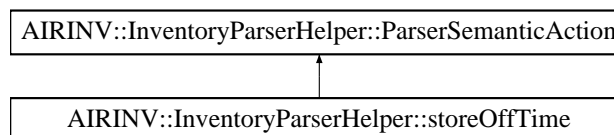
The documentation for this struct was generated from the following files:

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

## 10.176 AIRINV::InventoryParserHelper::storeOffTime Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeOffTime::



### Public Member Functions

- [storeOffTime](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.176.1 Detailed Description

Store the off time.

Definition at line 141 of file InventoryParserHelper.hpp.

## 10.176.2 Constructor & Destructor Documentation

### 10.176.2.1 AIRINV::InventoryParserHelper::storeOffTime::storeOffTime ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 242 of file InventoryParserHelper.cpp.

## 10.176.3 Member Function Documentation

### 10.176.3.1 void AIRINV::InventoryParserHelper::storeOffTime::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 247 of file InventoryParserHelper.cpp.

References AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_itLeg, AIRINV::FlightDateStruct::\_itSeconds, AIRINV::LegStruct::\_offTime, and AIRINV::FlightDateStruct::getTime().

## 10.176.4 Member Data Documentation

### 10.176.4.1 [FlightDateStruct&](#) AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFCClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::store-

SaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

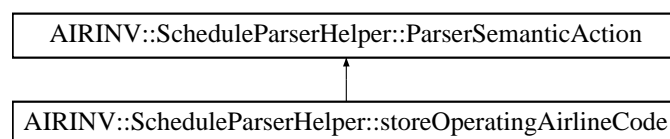
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.177 AIRINV::ScheduleParserHelper::storeOperatingAirlineCode Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeOperatingAirlineCode::



### Public Member Functions

- [storeOperatingAirlineCode](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.177.1 Detailed Description

Store the parsed operating airline code.

Definition at line 93 of file [ScheduleParserHelper.hpp](#).

#### 10.177.2 Constructor & Destructor Documentation

##### 10.177.2.1 AIRINV::ScheduleParserHelper::storeOperatingAirlineCode::storeOperatingAirlineCode ([FlightPeriodStruct](#) &)

Actor Constructor.



Definition at line 162 of file ScheduleParserHelper.cpp.

### 10.177.3 Member Function Documentation

#### 10.177.3.1 void AIRINV::ScheduleParserHelper::storeOperatingAirlineCode::operator() (iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 167 of file ScheduleParserHelper.cpp.

References AIRINV::LegStruct::\_airlineCode, AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, and AIRINV::FlightPeriodStruct::\_itLeg.

### 10.177.4 Member Data Documentation

#### 10.177.4.1 FlightPeriodStruct& AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flight- Period [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

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

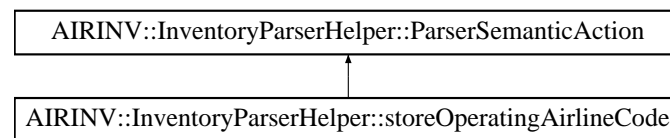
The documentation for this struct was generated from the following files:

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

## 10.178 AIRINV::InventoryParserHelper::storeOperatingAirlineCode Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeOperatingAirlineCode::



## Public Member Functions

- [storeOperatingAirlineCode](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

## Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

### 10.178.1 Detailed Description

Store the parsed operating airline code.

Definition at line 101 of file InventoryParserHelper.hpp.

### 10.178.2 Constructor & Destructor Documentation

#### 10.178.2.1 AIRINV::InventoryParserHelper::storeOperatingAirlineCode::storeOperatingAirlineCode ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 176 of file InventoryParserHelper.cpp.

### 10.178.3 Member Function Documentation

#### 10.178.3.1 void AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 181 of file InventoryParserHelper.cpp.

References [AIRINV::LegStruct::\\_airlineCode](#), [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), and [AIRINV::FlightDateStruct::\\_itLeg](#).

### 10.178.4 Member Data Documentation

#### 10.178.4.1 [FlightDateStruct](#)& [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#) [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by [AIRINV::InventoryParserHelper::doEndFlightDate::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFClasses::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFamilyCode::operator\(\)](#), and [AIRINV::InventoryParserHelper::storeRevenueAvailability::operator\(\)](#).

AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

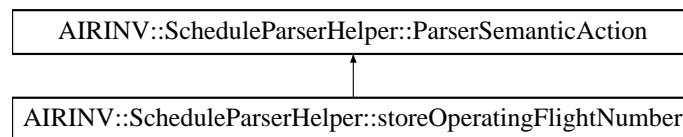
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.179 AIRINV::ScheduleParserHelper::storeOperatingFlightNumber Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeOperatingFlightNumber:



## Public Member Functions

- [storeOperatingFlightNumber](#) ([FlightPeriodStruct](#) &)
- void [operator\(\)](#) (unsigned int iNumber) const

## Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

### 10.179.1 Detailed Description

Store the parsed operating flight number.

Definition at line 101 of file `ScheduleParserHelper.hpp`.

### 10.179.2 Constructor & Destructor Documentation

#### 10.179.2.1 AIRINV::ScheduleParserHelper::storeOperatingFlightNumber::storeOperatingFlightNumber ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 178 of file `ScheduleParserHelper.cpp`.

### 10.179.3 Member Function Documentation

#### 10.179.3.1 void AIRINV::ScheduleParserHelper::storeOperatingFlightNumber::operator() (unsigned int iNumber) const

Actor Function (functor).

Definition at line 183 of file `ScheduleParserHelper.cpp`.

References `AIRINV::LegStruct::_flightNumber`, `AIRINV::ScheduleParserHelper::ParserSemanticAction::_flightPeriod`, and `AIRINV::FlightPeriodStruct::_itLeg`.

### 10.179.4 Member Data Documentation

#### 10.179.4.1 [FlightPeriodStruct](#)& AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [inherited]

Actor Context.

Definition at line 33 of file `ScheduleParserHelper.hpp`.

Referenced by `AIRINV::ScheduleParserHelper::doEndFlight::operator()`, `AIRINV::ScheduleParserHelper::storeFClasses::operator()`, `AIRINV::ScheduleParserHelper::storeFFDisutilityCurveKey::operator()`, `AIRINV::ScheduleParserHelper::storeFRAT5CurveKey::operator()`, `AIRINV::ScheduleParserHelper::storeFamilyCode::operator()`, `AIRINV::ScheduleParserHelper::storeClasses::operator()`, `AIRINV::ScheduleParserHelper::storeSegmentCabinCode::operator()`, `AIRINV::ScheduleParserHelper::storeSegmentOffPoint::operator()`, `AIRINV::ScheduleParserHelper::storeSegmentBoardingPoint::operator()`, `AIRINV::ScheduleParserHelper::storeSegmentSpecificity::operator()`, `AIRINV::ScheduleParserHelper::storeCapacity::operator()`, `AIRINV::ScheduleParserHelper::storeLegCabinCode::operator()`, `AIRINV::ScheduleParserHelper::storeElapsedTime::operator()`, and `AIRINV::ScheduleParserHelper::storeOffTime::operator()`.

AIRINV::ScheduleParserHelper::storeBoardingTime::operator(), operator(), AIRINV::ScheduleParserHelper::storeOperatingAirlineCode::operator(), AIRINV::ScheduleParserHelper::storeLegOffPoint::operator(), AIRINV::ScheduleParserHelper::storeLegBoardingPoint::operator(), AIRINV::ScheduleParserHelper::storeDow::operator(), AIRINV::ScheduleParserHelper::storeDateRangeEnd::operator(), AIRINV::ScheduleParserHelper::storeDateRangeStart::operator(), AIRINV::ScheduleParserHelper::storeFlightNumber::operator(), and AIRINV::ScheduleParserHelper::storeAirlineCode::operator().

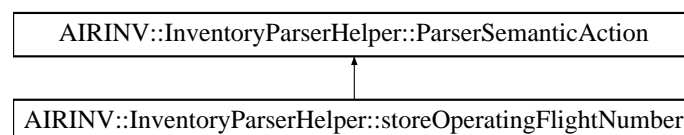
The documentation for this struct was generated from the following files:

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

## 10.180 AIRINV::InventoryParserHelper::storeOperatingFlightNumber Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeOperatingFlightNumber::



### Public Member Functions

- [storeOperatingFlightNumber](#) ([FlightDateStruct](#) &)
- [operator\(\)](#) (unsigned int iNumber) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.180.1 Detailed Description

Store the parsed operating flight number.

Definition at line 109 of file [InventoryParserHelper.hpp](#).

#### 10.180.2 Constructor & Destructor Documentation

##### 10.180.2.1 AIRINV::InventoryParserHelper::storeOperatingFlightNumber::storeOperatingFlightNumber ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 192 of file [InventoryParserHelper.cpp](#).

### 10.180.3 Member Function Documentation

#### 10.180.3.1 void AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator() (unsigned int *iNumber*) const

Actor Function (functor).

Definition at line 197 of file InventoryParserHelper.cpp.

References AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::LegStruct::\_flightNumber, and AIRINV::FlightDateStruct::\_itLeg.

### 10.180.4 Member Data Documentation

#### 10.180.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

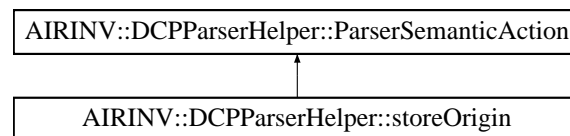
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.181 AIRINV::DCPParserHelper::storeOrigin Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::storeOrigin::



### Public Member Functions

- [storeOrigin](#) (DCPRuleStruct &)
- void [operator\(\)](#) (std::vector< char >, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

### Public Attributes

- DCPRuleStruct & [\\_DCPRule](#)

#### 10.181.1 Detailed Description

Store the parsed origin.

Definition at line 48 of file DCPParserHelper.hpp.

#### 10.181.2 Constructor & Destructor Documentation

##### 10.181.2.1 AIRINV::DCPParserHelper::storeOrigin::storeOrigin (DCPRuleStruct &)

Actor Constructor.

Definition at line 54 of file DCPParserHelper.cpp.

#### 10.181.3 Member Function Documentation

##### 10.181.3.1 void AIRINV::DCPParserHelper::storeOrigin::operator() (std::vector< char >, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

Actor Function (functor).

Definition at line 59 of file DCPParserHelper.cpp.

References AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule.

### 10.181.4 Member Data Documentation

#### 10.181.4.1 DCPRuleStruct& AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule [inherited]

Actor Context.

Definition at line 34 of file DCPParserHelper.hpp.

Referenced by AIRINV::DCPParserHelper::doEndDCP::operator(), AIRINV::DCPParserHelper::storeClass::operator(), AIRINV::DCPParserHelper::storeAirlineCode::operator(), AIRINV::DCPParserHelper::storeDCP::operator(), AIRINV::DCPParserHelper::storeMinimumStay::operator(), AIRINV::DCPParserHelper::storeNonRefundable::operator(), AIRINV::DCPParserHelper::storeChangeFees::operator(), AIRINV::DCPParserHelper::storeSaturdayStay::operator(), AIRINV::DCPParserHelper::storeAdvancePurchase::operator(), AIRINV::DCPParserHelper::storeChannel::operator(), AIRINV::DCPParserHelper::storeCabinCode::operator(), AIRINV::DCPParserHelper::storePOS::operator(), AIRINV::DCPParserHelper::storeEndRangeTime::operator(), AIRINV::DCPParserHelper::storeStartRangeTime::operator(), AIRINV::DCPParserHelper::storeDateRangeEnd::operator(), AIRINV::DCPParserHelper::storeDateRangeStart::operator(), AIRINV::DCPParserHelper::storeDestination::operator(), operator(), and AIRINV::DCPParserHelper::storeDCPID::operator().

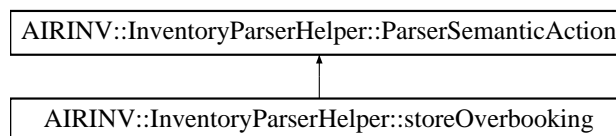
The documentation for this struct was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.182 AIRINV::InventoryParserHelper::storeOverbooking Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeOverbooking::



### Public Member Functions

- [storeOverbooking](#) ([FlightDateStruct](#) &)
- void [operator\(\)](#) (double iReal) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

### 10.182.1 Detailed Description

Store the parsed Overbooking percentage (at booking class level).

Definition at line 341 of file InventoryParserHelper.hpp.



## 10.182.2 Constructor & Destructor Documentation

### 10.182.2.1 AIRINV::InventoryParserHelper::storeOverbooking::storeOverbooking ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 623 of file InventoryParserHelper.cpp.

## 10.182.3 Member Function Documentation

### 10.182.3.1 void AIRINV::InventoryParserHelper::storeOverbooking::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 628 of file InventoryParserHelper.cpp.

References AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_itBookingClass, and AIRINV::BookingClassStruct::\_overbookingPercentage.

## 10.182.4 Member Data Documentation

### 10.182.4.1 [FlightDateStruct](#)& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFCClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParser-

Helper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(),  
 AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParser-  
 Helper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlight-  
 Number::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(),  
 AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::store-  
 LegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(),  
 AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParser-  
 Helper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(),  
 AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParser-  
 Helper::storeSnapshotDate::operator().

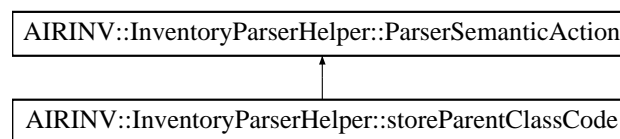
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.183 AIRINV::InventoryParserHelper::storeParentClassCode Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeParentClassCode::



### Public Member Functions

- [storeParentClassCode](#) ([FlightDateStruct](#) &)
- [operator\(\)](#) (char iChar) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.183.1 Detailed Description

Store the parsed class code of the parent sub-class.

Definition at line 293 of file [InventoryParserHelper.hpp](#).

#### 10.183.2 Constructor & Destructor Documentation

##### 10.183.2.1 AIRINV::InventoryParserHelper::storeParentClassCode::storeParentClassCode ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 555 of file [InventoryParserHelper.cpp](#).

### 10.183.3 Member Function Documentation

#### 10.183.3.1 void AIRINV::InventoryParserHelper::storeParentClassCode::operator() (char iChar) const

Actor Function (functor).

Definition at line 560 of file InventoryParserHelper.cpp.

References AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_itBookingClass, and AIRINV::BookingClassStruct::\_parentClassCode.

### 10.183.4 Member Data Documentation

#### 10.183.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParser-

Helper::storeSnapshotDate::operator()().

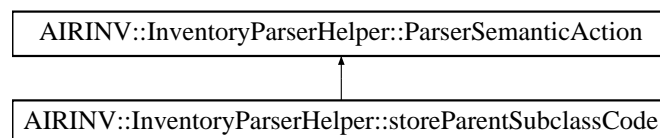
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.184 AIRINV::InventoryParserHelper::storeParentSubclassCode Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeParentSubclassCode::



### Public Member Functions

- [storeParentSubclassCode](#) ([FlightDateStruct](#) &)
- void [operator\(\)](#) (unsigned int iNumber) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.184.1 Detailed Description

Store the parsed sub-class code of the parent sub-class.

Definition at line 301 of file InventoryParserHelper.hpp.

#### 10.184.2 Constructor & Destructor Documentation

##### 10.184.2.1 AIRINV::InventoryParserHelper::storeParentSubclassCode::storeParentSubclassCode ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 567 of file InventoryParserHelper.cpp.

#### 10.184.3 Member Function Documentation

##### 10.184.3.1 void AIRINV::InventoryParserHelper::storeParentSubclassCode::operator() (unsigned int iNumber) const

Actor Function (functor).

Definition at line 572 of file InventoryParserHelper.cpp.

References AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_itBookingClass, and AIRINV::BookingClassStruct::\_parentSubclassCode.

#### 10.184.4 Member Data Documentation

##### 10.184.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

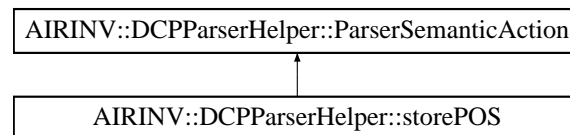
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.185 AIRINV::DCPParserHelper::storePOS Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::storePOS::



### Public Member Functions

- [storePOS](#) (DCPRuleStruct &)
- void [operator\(\)](#) (std::vector< char >, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

### Public Attributes

- DCPRuleStruct & [\\_DCPRule](#)

#### 10.185.1 Detailed Description

Store the parsed customer position.

Definition at line 108 of file DCPParserHelper.hpp.

#### 10.185.2 Constructor & Destructor Documentation

##### 10.185.2.1 AIRINV::DCPParserHelper::storePOS::storePOS (DCPRuleStruct &)

Actor Constructor.

Definition at line 150 of file DCPParserHelper.cpp.

#### 10.185.3 Member Function Documentation

##### 10.185.3.1 void AIRINV::DCPParserHelper::storePOS::operator() (std::vector< char >, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

Actor Function (functor).

Definition at line 155 of file DCPParserHelper.cpp.

References AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule.

#### 10.185.4 Member Data Documentation

##### 10.185.4.1 DCPRuleStruct& [AIRINV::DCPParserHelper::ParserSemanticAction::\\_DCPRule](#) [inherited]

Actor Context.

Definition at line 34 of file DCPParserHelper.hpp.

Referenced by AIRINV::DCPParserHelper::doEndDCP::operator(), AIRINV::DCPParserHelper::storeClass::operator(), AIRINV::DCPParserHelper::storeAirlineCode::operator(), AIRINV::DCPParserHelper::storeDCP::operator(), AIRINV::DCPParserHelper::storeMinimumStay::operator(), AIRINV::DCPParserHelper::storeNonRefundable::operator(), AIRINV::DCPParserHelper::storeChangeFees::operator(), AIRINV::DCPParserHelper::storeSaturdayStay::operator(), AIRINV::DCPParserHelper::storeAdvancePurchase::operator(), AIRINV::DCPParserHelper::storeChannel::operator(), AIRINV::DCPParserHelper::storeCabinCode::operator(), AIRINV::DCPParserHelper::storeEndRangeTime::operator(), AIRINV::DCPParserHelper::storeStartRangeTime::operator(), AIRINV::DCPParserHelper::storeDateRangeEnd::operator(), AIRINV::DCPParserHelper::storeDateRangeStart::operator(), AIRINV::DCPParserHelper::storeDestination::operator(), AIRINV::DCPParserHelper::storeOrigin::operator(), and AIRINV::DCPParserHelper::storeDCPid::operator().

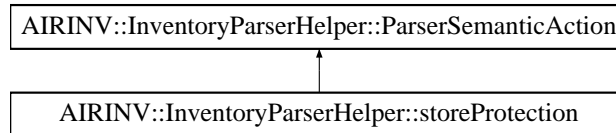
The documentation for this struct was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.186 AIRINV::InventoryParserHelper::storeProtection Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeProtection::



### Public Member Functions

- [storeProtection](#) ([FlightDateStruct](#) &)
- void [operator\(\)](#) (double iReal) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.186.1 Detailed Description

Store the parsed protection (at booking class level).

Definition at line 317 of file InventoryParserHelper.hpp.

#### 10.186.2 Constructor & Destructor Documentation

##### 10.186.2.1 AIRINV::InventoryParserHelper::storeProtection::storeProtection ([FlightDateStruct](#) &)



Actor Constructor.

Definition at line 590 of file InventoryParserHelper.cpp.

### 10.186.3 Member Function Documentation

#### 10.186.3.1 void AIRINV::InventoryParserHelper::storeProtection::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 595 of file InventoryParserHelper.cpp.

References AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_itBookingClass, and AIRINV::BookingClassStruct::\_protection.

### 10.186.4 Member Data Documentation

#### 10.186.4.1 [FlightDateStruct](#)& [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#) [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::store-



LegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

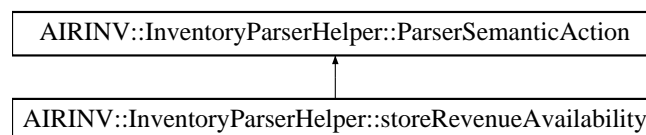
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.187 AIRINV::InventoryParserHelper::storeRevenueAvailability Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeRevenueAvailability::



### Public Member Functions

- [storeRevenueAvailability](#) ([FlightDateStruct](#) &)
- void [operator\(\)](#) (double iReal) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.187.1 Detailed Description

Store the parsed number of net revenue availability (at booking class level).

Definition at line 417 of file InventoryParserHelper.hpp.

#### 10.187.2 Constructor & Destructor Documentation

##### 10.187.2.1 AIRINV::InventoryParserHelper::storeRevenueAvailability::storeRevenueAvailability ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 726 of file InventoryParserHelper.cpp.

### 10.187.3 Member Function Documentation

#### 10.187.3.1 void AIRINV::InventoryParserHelper::storeRevenueAvailability::operator() (double *i-Real*) const

Actor Function (functor).

Definition at line 731 of file InventoryParserHelper.cpp.

References AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_itBookingClass, and AIRINV::BookingClassStruct::\_netRevenueAvailability.

### 10.187.4 Member Data Documentation

#### 10.187.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

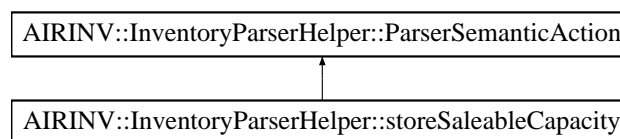
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.188 AIRINV::InventoryParserHelper::storeSaleableCapacity Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeSaleableCapacity::



### Public Member Functions

- [storeSaleableCapacity](#) ([FlightDateStruct](#) &)
- [operator\(\)](#) (double iReal) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.188.1 Detailed Description

Store the parsed saleable capacity.

Definition at line 157 of file [InventoryParserHelper.hpp](#).

#### 10.188.2 Constructor & Destructor Documentation

##### 10.188.2.1 AIRINV::InventoryParserHelper::storeSaleableCapacity::storeSaleableCapacity ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 284 of file [InventoryParserHelper.cpp](#).

#### 10.188.3 Member Function Documentation

##### 10.188.3.1 void AIRINV::InventoryParserHelper::storeSaleableCapacity::operator() (double *i-Real*) const

Actor Function (functor).

Definition at line 289 of file [InventoryParserHelper.cpp](#).

References [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), [AIRINV::FlightDateStruct::\\_itLegCabin](#), and [AIRINV::LegCabinStruct::\\_saleableCapacity](#).

#### 10.188.4 Member Data Documentation

##### 10.188.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flight-Date [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvaibility::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

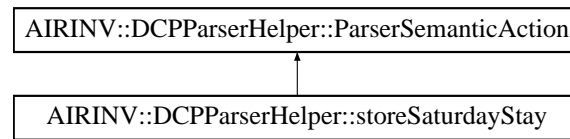
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.189 AIRINV::DCPParserHelper::storeSaturdayStay Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::storeSaturdayStay::



## Public Member Functions

- [storeSaturdayStay](#) (DCPRuleStruct &)
- void [operator\(\)](#) (char, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

## Public Attributes

- DCPRuleStruct & [\\_DCPRule](#)

### 10.189.1 Detailed Description

Store the parsed saturday night.

Definition at line 148 of file DCPParserHelper.hpp.

### 10.189.2 Constructor & Destructor Documentation

#### 10.189.2.1 AIRINV::DCPParserHelper::storeSaturdayStay::storeSaturdayStay (DCPRuleStruct &)

Actor Constructor.

Definition at line 223 of file DCPParserHelper.cpp.

### 10.189.3 Member Function Documentation

#### 10.189.3.1 void AIRINV::DCPParserHelper::storeSaturdayStay::operator() (char, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

Actor Function (functor).

Definition at line 228 of file DCPParserHelper.cpp.

References AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule.

### 10.189.4 Member Data Documentation

#### 10.189.4.1 DCPRuleStruct& [AIRINV::DCPParserHelper::ParserSemanticAction::\\_DCPRule](#) [inherited]

Actor Context.

Definition at line 34 of file DCPParserHelper.hpp.

Referenced by AIRINV::DCPParserHelper::doEndDCP::operator>(), AIRINV::DCPParserHelper::storeClass::operator>(), AIRINV::DCPParserHelper::storeAirlineCode::operator>(), AIRINV::DCPParserHelper::storeDCP::operator>(), AIRINV::DCPParserHelper::storeMinimumStay::operator>(),

AIRINV::DCPParserHelper::storeNonRefundable::operator(), AIRINV::DCPParserHelper::storeChangeFees::operator(), operator(), AIRINV::DCPParserHelper::storeAdvancePurchase::operator(), AIRINV::DCPParserHelper::storeChannel::operator(), AIRINV::DCPParserHelper::storeCabinCode::operator(), AIRINV::DCPParserHelper::storePOS::operator(), AIRINV::DCPParserHelper::storeEndRangeTime::operator(), AIRINV::DCPParserHelper::storeStartRangeTime::operator(), AIRINV::DCPParserHelper::storeDateRangeEnd::operator(), AIRINV::DCPParserHelper::storeDateRangeStart::operator(), AIRINV::DCPParserHelper::storeDestination::operator(), AIRINV::DCPParserHelper::storeOrigin::operator(), and AIRINV::DCPParserHelper::storeDCPId::operator().

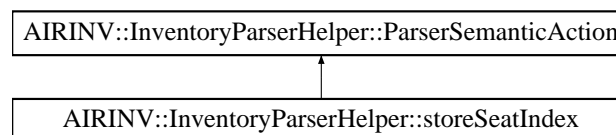
The documentation for this struct was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.190 AIRINV::InventoryParserHelper::storeSeatIndex Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeSeatIndex::



### Public Member Functions

- [storeSeatIndex](#) ([FlightDateStruct](#) &)
- [operator\(\)](#) (double iReal) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.190.1 Detailed Description

Store the parsed leg-cabin seat index.

Definition at line 237 of file [InventoryParserHelper.hpp](#).

#### 10.190.2 Constructor & Destructor Documentation

##### 10.190.2.1 AIRINV::InventoryParserHelper::storeSeatIndex::storeSeatIndex ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 403 of file [InventoryParserHelper.cpp](#).

### 10.190.3 Member Function Documentation

#### 10.190.3.1 void AIRINV::InventoryParserHelper::storeSeatIndex::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 408 of file InventoryParserHelper.cpp.

References AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_itBucket, and AIRINV::BucketStruct::\_seatIndex.

### 10.190.4 Member Data Documentation

#### 10.190.4.1 [FlightDateStruct](#)& [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#) [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

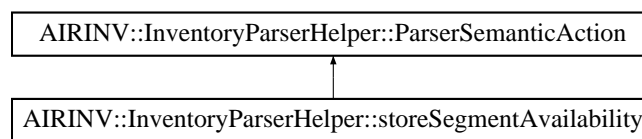
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.191 AIRINV::InventoryParserHelper::storeSegmentAvailability Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeSegmentAvailability::



### Public Member Functions

- [storeSegmentAvailability](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) (double iReal) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.191.1 Detailed Description

Store the parsed number of segment availability (at booking class level).

Definition at line 408 of file [InventoryParserHelper.hpp](#).

#### 10.191.2 Constructor & Destructor Documentation

##### 10.191.2.1 AIRINV::InventoryParserHelper::storeSegmentAvailability::storeSegmentAvailability ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 714 of file [InventoryParserHelper.cpp](#).

#### 10.191.3 Member Function Documentation

##### 10.191.3.1 void AIRINV::InventoryParserHelper::storeSegmentAvailability::operator() (double *i-Real*) const

Actor Function (functor).

Definition at line 719 of file [InventoryParserHelper.cpp](#).

References [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), [AIRINV::FlightDateStruct::\\_itBookingClass](#), and [AIRINV::BookingClassStruct::\\_segmentAvailability](#).



### 10.191.4 Member Data Documentation

#### 10.191.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvaibility::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

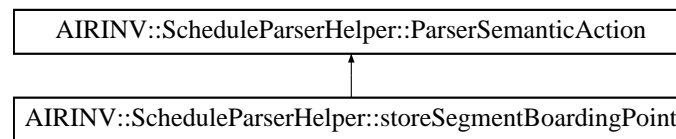
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.192 AIRINV::ScheduleParserHelper::storeSegmentBoardingPoint Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeSegmentBoardingPoint::



## Public Member Functions

- [storeSegmentBoardingPoint](#) ([FlightPeriodStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

## Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

### 10.192.1 Detailed Description

Store the parsed segment boarding point.

Definition at line 160 of file ScheduleParserHelper.hpp.

### 10.192.2 Constructor & Destructor Documentation

#### 10.192.2.1 AIRINV::ScheduleParserHelper::storeSegmentBoardingPoint::storeSegmentBoardingPoint ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 307 of file ScheduleParserHelper.cpp.

### 10.192.3 Member Function Documentation

#### 10.192.3.1 void AIRINV::ScheduleParserHelper::storeSegmentBoardingPoint::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 312 of file ScheduleParserHelper.cpp.

References [AIRINV::SegmentStruct::\\_boardingPoint](#), [AIRINV::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#), and [AIRINV::FlightPeriodStruct::\\_itSegment](#).

### 10.192.4 Member Data Documentation

#### 10.192.4.1 [FlightPeriodStruct](#)& AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

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

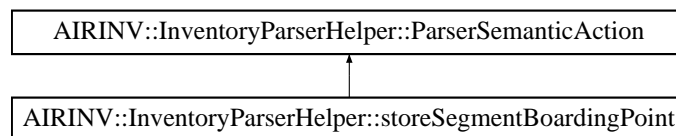
The documentation for this struct was generated from the following files:

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

## 10.193 AIRINV::InventoryParserHelper::storeSegmentBoardingPoint Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::



### Public Member Functions

- [storeSegmentBoardingPoint](#) ([FlightDateStruct](#) &)
- [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.193.1 Detailed Description

Store the parsed segment boarding point.

Definition at line 245 of file [InventoryParserHelper.hpp](#).

## 10.193.2 Constructor & Destructor Documentation

### 10.193.2.1 AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::storeSegmentBoardingPoint (FlightDateStruct &)

Actor Constructor.

Definition at line 415 of file InventoryParserHelper.cpp.

## 10.193.3 Member Function Documentation

### 10.193.3.1 void AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator()(iterator\_t iStr, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 420 of file InventoryParserHelper.cpp.

References AIRINV::SegmentStruct::\_boardingPoint, AIRINV::LegCabinStruct::\_bucketList, AIRINV::LegCabinStruct::\_cabinCode, AIRINV::SegmentStruct::\_cabinList, AIRINV::LegStruct::\_cabinList, AIRINV::BookingClassStruct::\_classCode, AIRINV::FareFamilyStruct::\_classList, AIRINV::SegmentCabinStruct::\_fareFamilies, AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_itBookingClass, AIRINV::FlightDateStruct::\_itBucket, AIRINV::SegmentCabinStruct::\_itFareFamily, AIRINV::FlightDateStruct::\_itLeg, AIRINV::FlightDateStruct::\_itLegCabin, AIRINV::FlightDateStruct::\_itSegment, AIRINV::FlightDateStruct::\_itSegmentCabin, AIRINV::FlightDateStruct::\_legList, and AIRINV::FlightDateStruct::\_segmentList.

## 10.193.4 Member Data Documentation

### 10.193.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(),

AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

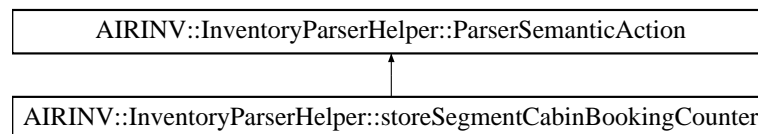
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.194 AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::



### Public Member Functions

- [storeSegmentCabinBookingCounter](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) (double iReal) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.194.1 Detailed Description

Store the parsed segment cabin number of bookings.

Definition at line 269 of file [InventoryParserHelper.hpp](#).

## 10.194.2 Constructor & Destructor Documentation

### 10.194.2.1 AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::storeSegmentCabinBookingCounter ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 513 of file InventoryParserHelper.cpp.

## 10.194.3 Member Function Documentation

### 10.194.3.1 void AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator() (double *iReal*) const

Actor Function (functor).

Definition at line 518 of file InventoryParserHelper.cpp.

References AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_itSegmentCabin, and AIRINV::SegmentCabinStruct::\_nbOfBookings.

## 10.194.4 Member Data Documentation

### 10.194.4.1 [FlightDateStruct](#) & AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFCClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::store-

OffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(),  
 AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParser-  
 Helper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperating-  
 AirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(),  
 AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParser-  
 Helper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightType-  
 Code::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::Inventory-  
 ParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirline-  
 Code::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator()).

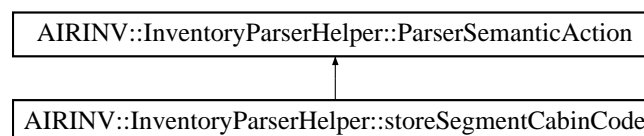
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.195 AIRINV::InventoryParserHelper::storeSegmentCabinCode Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeSegmentCabinCode::



### Public Member Functions

- [storeSegmentCabinCode](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) (char iChar) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

#### 10.195.1 Detailed Description

Store the parsed segment cabin code.

Definition at line 261 of file [InventoryParserHelper.hpp](#).

#### 10.195.2 Constructor & Destructor Documentation

##### 10.195.2.1 AIRINV::InventoryParserHelper::storeSegmentCabinCode::storeSegmentCabinCode ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 478 of file [InventoryParserHelper.cpp](#).



### 10.195.3 Member Function Documentation

#### 10.195.3.1 void AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator() (char *i-Char*) const

Actor Function (functor).

Definition at line 483 of file InventoryParserHelper.cpp.

References AIRINV::SegmentCabinStruct::\_cabinCode, AIRINV::SegmentStruct::\_cabinList, AIRINV::BookingClassStruct::\_classCode, AIRINV::FareFamilyStruct::\_classList, AIRINV::SegmentCabinStruct::\_fareFamilies, AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_itBookingClass, AIRINV::SegmentCabinStruct::\_itFareFamily, AIRINV::FlightDateStruct::\_itSegment, and AIRINV::FlightDateStruct::\_itSegmentCabin.

### 10.195.4 Member Data Documentation

#### 10.195.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(),



AIRINV::InventoryParserHelper::storeFlightTypeCode::operator>(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

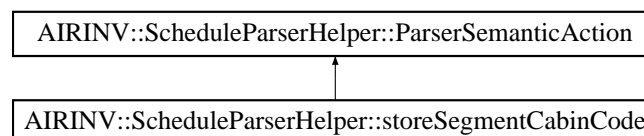
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.196 AIRINV::ScheduleParserHelper::storeSegmentCabinCode Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeSegmentCabinCode::



### Public Member Functions

- [storeSegmentCabinCode](#) ([FlightPeriodStruct](#) &)
- [operator\(\)](#) (char iChar) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.196.1 Detailed Description

Store the parsed segment cabin code.

Definition at line 176 of file [ScheduleParserHelper.hpp](#).

#### 10.196.2 Constructor & Destructor Documentation

##### 10.196.2.1 AIRINV::ScheduleParserHelper::storeSegmentCabinCode::storeSegmentCabinCode ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 333 of file [ScheduleParserHelper.cpp](#).

#### 10.196.3 Member Function Documentation

##### 10.196.3.1 void AIRINV::ScheduleParserHelper::storeSegmentCabinCode::operator() (char *i-Char*) const

Actor Function (functor).

Definition at line 338 of file ScheduleParserHelper.cpp.

References AIRINV::SegmentCabinStruct::\_cabinCode, AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, and AIRINV::FlightPeriodStruct::\_itSegmentCabin.

#### 10.196.4 Member Data Documentation

##### 10.196.4.1 FlightPeriodStruct& AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flight-Period [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

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

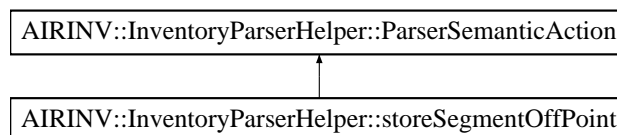
The documentation for this struct was generated from the following files:

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

## 10.197 AIRINV::InventoryParserHelper::storeSegmentOffPoint Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeSegmentOffPoint::



#### Public Member Functions

- [storeSegmentOffPoint](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

**Public Attributes**

- [FlightDateStruct](#) & [\\_flightDate](#)

**10.197.1 Detailed Description**

Store the parsed segment off point.

Definition at line 253 of file InventoryParserHelper.hpp.

**10.197.2 Constructor & Destructor Documentation****10.197.2.1 AIRINV::InventoryParserHelper::storeSegmentOffPoint::storeSegmentOffPoint([FlightDateStruct](#) &)**

Actor Constructor.

Definition at line 464 of file InventoryParserHelper.cpp.

**10.197.3 Member Function Documentation****10.197.3.1 void AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator()([iterator\\_t i-Str](#), [iterator\\_t iStrEnd](#)) const**

Actor Function (functor).

Definition at line 469 of file InventoryParserHelper.cpp.

References [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), [AIRINV::FlightDateStruct::\\_itSegment](#), and [AIRINV::SegmentStruct::\\_offPoint](#).

**10.197.4 Member Data Documentation****10.197.4.1 [FlightDateStruct&](#) [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#) [inherited]**

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by [AIRINV::InventoryParserHelper::doEndFlightDate::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFCClasses::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFamilyCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeRevenueAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeSegmentAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassETB::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeOverbooking::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNoShow::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNego::operator\(\)](#), [AIRINV::InventoryParserHelper::storeProtection::operator\(\)](#), [AIRINV::InventoryParserHelper::storeCumulatedProtection::operator\(\)](#), [AIRINV::InventoryParserHelper::storeParentSubclassCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeParentClassCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeSubclassCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassCode::operator\(\)](#),

AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

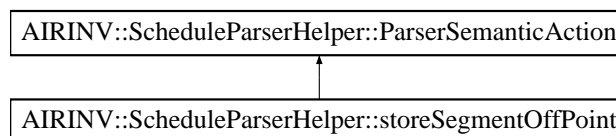
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.198 AIRINV::ScheduleParserHelper::storeSegmentOffPoint Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeSegmentOffPoint:



### Public Member Functions

- [storeSegmentOffPoint](#) ([FlightPeriodStruct](#) &)
- [operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

#### 10.198.1 Detailed Description

Store the parsed segment off point.

Definition at line 168 of file [ScheduleParserHelper.hpp](#).

## 10.198.2 Constructor & Destructor Documentation

### 10.198.2.1 AIRINV::ScheduleParserHelper::storeSegmentOffPoint::storeSegmentOffPoint (FlightPeriodStruct &)

Actor Constructor.

Definition at line 320 of file ScheduleParserHelper.cpp.

## 10.198.3 Member Function Documentation

### 10.198.3.1 void AIRINV::ScheduleParserHelper::storeSegmentOffPoint::operator() (iterator\_t i-Str, iterator\_t iStrEnd) const

Actor Function (functor).

Definition at line 325 of file ScheduleParserHelper.cpp.

References AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flightPeriod, AIRINV::FlightPeriodStruct::\_itSegment, and AIRINV::SegmentStruct::\_offPoint.

## 10.198.4 Member Data Documentation

### 10.198.4.1 FlightPeriodStruct& AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flight-Period [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

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

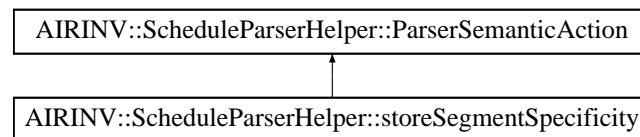
The documentation for this struct was generated from the following files:

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

## 10.199 AIRINV::ScheduleParserHelper::storeSegmentSpecificity Struct Reference

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

Inheritance diagram for AIRINV::ScheduleParserHelper::storeSegmentSpecificity::



### Public Member Functions

- [storeSegmentSpecificity](#) ([FlightPeriodStruct](#) &)
- void [operator\(\)](#) (char iChar) const

### Public Attributes

- [FlightPeriodStruct](#) & [\\_flightPeriod](#)

### 10.199.1 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.

### 10.199.2 Constructor & Destructor Documentation

#### 10.199.2.1 AIRINV::ScheduleParserHelper::storeSegmentSpecificity::storeSegmentSpecificity ([FlightPeriodStruct](#) &)

Actor Constructor.

Definition at line 281 of file ScheduleParserHelper.cpp.

### 10.199.3 Member Function Documentation

#### 10.199.3.1 void AIRINV::ScheduleParserHelper::storeSegmentSpecificity::operator() (char *iChar*) const

Actor Function (functor).

Definition at line 286 of file ScheduleParserHelper.cpp.

References [AIRINV::FlightPeriodStruct::\\_airportList](#), [AIRINV::FlightPeriodStruct::\\_airportOrderedList](#), [AIRINV::FlightPeriodStruct::\\_areSegmentDefinitionsSpecific](#), [AIRINV::ScheduleParserHelper::ParserSemanticAction::\\_flightPeriod](#), and [AIRINV::FlightPeriodStruct::buildSegments\(\)](#).

#### 10.199.4 Member Data Documentation

##### 10.199.4.1 FlightPeriodStruct& AIRINV::ScheduleParserHelper::ParserSemanticAction::\_flight-Period [inherited]

Actor Context.

Definition at line 33 of file ScheduleParserHelper.hpp.

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

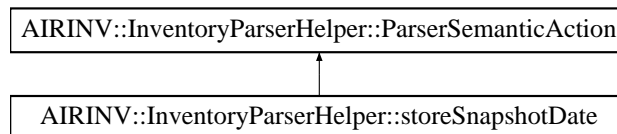
The documentation for this struct was generated from the following files:

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

## 10.200 AIRINV::InventoryParserHelper::storeSnapshotDate Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeSnapshotDate::



### Public Member Functions

- [storeSnapshotDate](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

### 10.200.1 Detailed Description

Store the snapshot date.

Definition at line 37 of file InventoryParserHelper.hpp.

### 10.200.2 Constructor & Destructor Documentation

#### 10.200.2.1 AIRINV::InventoryParserHelper::storeSnapshotDate::storeSnapshotDate ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 32 of file InventoryParserHelper.cpp.

### 10.200.3 Member Function Documentation

#### 10.200.3.1 void AIRINV::InventoryParserHelper::storeSnapshotDate::operator() ([iterator\\_t](#) iStr, [iterator\\_t](#) iStrEnd) const

Actor Function (functor).

Definition at line 37 of file InventoryParserHelper.cpp.

References AIRINV::FlightDateStruct::\_flightDate, AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, and AIRINV::FlightDateStruct::getDate().

### 10.200.4 Member Data Documentation

#### 10.200.4.1 [FlightDateStruct&](#) AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFCClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(),



AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and operator().

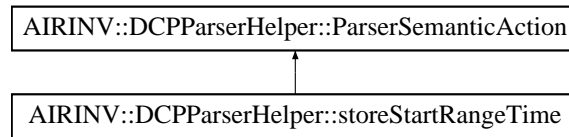
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.201 AIRINV::DCPParserHelper::storeStartRangeTime Struct Reference

```
#include <airinv/command/vault/DCPParserHelper.hpp>
```

Inheritance diagram for AIRINV::DCPParserHelper::storeStartRangeTime::



### Public Member Functions

- [storeStartRangeTime](#) (DCPRuleStruct &)
- void [operator\(\)](#) (boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

### Public Attributes

- DCPRuleStruct & [\\_DCPRule](#)

#### 10.201.1 Detailed Description

Store the parsed start range time.

Definition at line 88 of file DCPParserHelper.hpp.

### 10.201.2 Constructor & Destructor Documentation

#### 10.201.2.1 AIRINV::DCPParserHelper::storeStartRangeTime::storeStartRangeTime (DCPRuleStruct &)

Actor Constructor.

Definition at line 116 of file DCPParserHelper.cpp.

### 10.201.3 Member Function Documentation

#### 10.201.3.1 void AIRINV::DCPParserHelper::storeStartRangeTime::operator() (boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type, boost::spirit::qi::unused\_type) const

Actor Function (functor).

Definition at line 121 of file DCPParserHelper.cpp.

References AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule.

### 10.201.4 Member Data Documentation

#### 10.201.4.1 DCPRuleStruct& AIRINV::DCPParserHelper::ParserSemanticAction::\_DCPRule [inherited]

Actor Context.

Definition at line 34 of file DCPParserHelper.hpp.

Referenced by AIRINV::DCPParserHelper::doEndDCP::operator(), AIRINV::DCPParserHelper::storeClass::operator(), AIRINV::DCPParserHelper::storeAirlineCode::operator(), AIRINV::DCPParserHelper::storeDCP::operator(), AIRINV::DCPParserHelper::storeMinimumStay::operator(), AIRINV::DCPParserHelper::storeNonRefundable::operator(), AIRINV::DCPParserHelper::storeChangeFees::operator(), AIRINV::DCPParserHelper::storeSaturdayStay::operator(), AIRINV::DCPParserHelper::storeAdvancePurchase::operator(), AIRINV::DCPParserHelper::storeChannel::operator(), AIRINV::DCPParserHelper::storeCabinCode::operator(), AIRINV::DCPParserHelper::storePOS::operator(), AIRINV::DCPParserHelper::storeEndRangeTime::operator(), operator(), AIRINV::DCPParserHelper::storeDateRangeEnd::operator(), AIRINV::DCPParserHelper::storeDateRangeStart::operator(), AIRINV::DCPParserHelper::storeDestination::operator(), AIRINV::DCPParserHelper::storeOrigin::operator(), and AIRINV::DCPParserHelper::storeDCPId::operator().

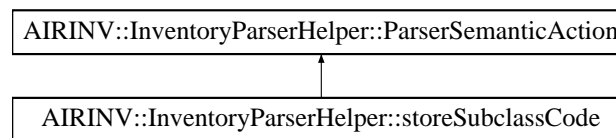
The documentation for this struct was generated from the following files:

- [airinv/command/vault/DCPParserHelper.hpp](#)
- [airinv/command/vault/DCPParserHelper.cpp](#)

## 10.202 AIRINV::InventoryParserHelper::storeSubclassCode Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeSubclassCode::



## Public Member Functions

- [storeSubclassCode](#) ([FlightDateStruct](#) &)
- void [operator\(\)](#) (unsigned int iNumber) const

## Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

### 10.202.1 Detailed Description

Store the parsed sub-class code.

Definition at line 285 of file InventoryParserHelper.hpp.

### 10.202.2 Constructor & Destructor Documentation

#### 10.202.2.1 AIRINV::InventoryParserHelper::storeSubclassCode::storeSubclassCode ([FlightDateStruct](#) &)

Actor Constructor.

Definition at line 543 of file InventoryParserHelper.cpp.

### 10.202.3 Member Function Documentation

#### 10.202.3.1 void AIRINV::InventoryParserHelper::storeSubclassCode::operator() (unsigned int *iNumber*) const

Actor Function (functor).

Definition at line 548 of file InventoryParserHelper.cpp.

References [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), [AIRINV::FlightDateStruct::\\_itBookingClass](#), and [AIRINV::BookingClassStruct::\\_subclassCode](#).

### 10.202.4 Member Data Documentation

#### 10.202.4.1 [FlightDateStruct](#)& [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#) [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by [AIRINV::InventoryParserHelper::doEndFlightDate::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFClasses::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFamilyCode::operator\(\)](#), and [AIRINV::InventoryParserHelper::storeRevenueAvailability::operator\(\)](#).

AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::storeBucketAvailability::operator(), AIRINV::InventoryParserHelper::storeYieldUpperRange::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(), AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

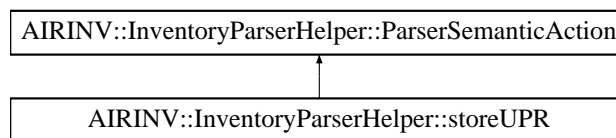
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.203 AIRINV::InventoryParserHelper::storeUPR Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeUPR::



**Public Member Functions**

- [storeUPR](#) ([FlightDateStruct](#) &)
- void [operator\(\)](#) (double *iReal*) const

**Public Attributes**

- [FlightDateStruct](#) & [\\_flightDate](#)

**10.203.1 Detailed Description**

Store the parsed Unsold Protected (UPR).

Definition at line 173 of file InventoryParserHelper.hpp.

**10.203.2 Constructor & Destructor Documentation****10.203.2.1 AIRINV::InventoryParserHelper::storeUPR::storeUPR ([FlightDateStruct](#) &)**

Actor Constructor.

Definition at line 306 of file InventoryParserHelper.cpp.

**10.203.3 Member Function Documentation****10.203.3.1 void AIRINV::InventoryParserHelper::storeUPR::operator() (double *iReal*) const**

Actor Function (functor).

Definition at line 311 of file InventoryParserHelper.cpp.

References [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#), [AIRINV::FlightDateStruct::\\_itLegCabin](#), and [AIRINV::LegCabinStruct::\\_upr](#).

**10.203.4 Member Data Documentation****10.203.4.1 [FlightDateStruct](#)& [AIRINV::InventoryParserHelper::ParserSemanticAction::\\_flightDate](#) [inherited]**

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by [AIRINV::InventoryParserHelper::doEndFlightDate::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFClasses::operator\(\)](#), [AIRINV::InventoryParserHelper::storeFamilyCode::operator\(\)](#), [AIRINV::InventoryParserHelper::storeRevenueAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeSegmentAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassAvailability::operator\(\)](#), [AIRINV::InventoryParserHelper::storeClassETB::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNbOfBkgs::operator\(\)](#), [AIRINV::InventoryParserHelper::storeOverbooking::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNoShow::operator\(\)](#), [AIRINV::InventoryParserHelper::storeNego::operator\(\)](#), [AIRINV::InventoryParserHelper::storeProtection::operator\(\)](#), and [AIRINV::InventoryParserHelper::storeCumulated](#)

Protection::operator>(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator>(),  
 AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::Inventory-  
 ParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClass-  
 Code::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(),  
 AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParser-  
 Helper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoarding-  
 Point::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::Inventory-  
 ParserHelper::storeBucketAvaibility::operator(), AIRINV::InventoryParserHelper::storeYieldUpper-  
 Range::operator(), AIRINV::InventoryParserHelper::storeETB::operator(), AIRINV::Inventory-  
 ParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::storeGAV::operator(),  
 AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParserHelper::store-  
 BookingCounter::operator(), operator(), AIRINV::InventoryParserHelper::storeAU::operator(),  
 AIRINV::InventoryParserHelper::storeSaleableCapacity::operator(), AIRINV::Inventory-  
 ParserHelper::storeLegCabinCode::operator(), AIRINV::InventoryParserHelper::storeOff-  
 Time::operator(), AIRINV::InventoryParserHelper::storeOffDate::operator(), AIRINV::Inventory-  
 ParserHelper::storeBoardingTime::operator(), AIRINV::InventoryParserHelper::storeBoarding-  
 Date::operator(), AIRINV::InventoryParserHelper::storeOperatingFlightNumber::operator(),  
 AIRINV::InventoryParserHelper::storeOperatingAirlineCode::operator(), AIRINV::Inventory-  
 ParserHelper::storeLegOffPoint::operator(), AIRINV::InventoryParserHelper::storeLegBoarding-  
 Point::operator(), AIRINV::InventoryParserHelper::storeFlightVisibilityCode::operator(),  
 AIRINV::InventoryParserHelper::storeFlightTypeCode::operator(), AIRINV::InventoryParser-  
 Helper::storeFlightDate::operator(), AIRINV::InventoryParserHelper::storeFlightNumber::operator(),  
 AIRINV::InventoryParserHelper::storeAirlineCode::operator(), and AIRINV::InventoryParser-  
 Helper::storeSnapshotDate::operator().

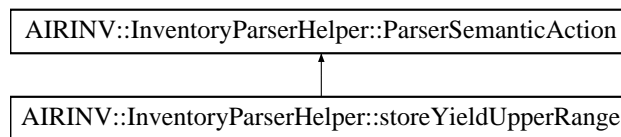
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.204 AIRINV::InventoryParserHelper::storeYieldUpperRange Struct Reference

```
#include <airinv/command/InventoryParserHelper.hpp>
```

Inheritance diagram for AIRINV::InventoryParserHelper::storeYieldUpperRange::



### Public Member Functions

- [storeYieldUpperRange](#) ([FlightDateStruct](#) &)
- [void operator\(\)](#) (double iReal) const

### Public Attributes

- [FlightDateStruct](#) & [\\_flightDate](#)

### 10.204.1 Detailed Description

Store the parsed Yield Upper Range value.

Definition at line 221 of file InventoryParserHelper.hpp.

### 10.204.2 Constructor & Destructor Documentation

#### 10.204.2.1 AIRINV::InventoryParserHelper::storeYieldUpperRange::storeYieldUpperRange (FlightDateStruct &)

Actor Constructor.

Definition at line 372 of file InventoryParserHelper.cpp.

### 10.204.3 Member Function Documentation

#### 10.204.3.1 void AIRINV::InventoryParserHelper::storeYieldUpperRange::operator() (double i-Real) const

Actor Function (functor).

Definition at line 377 of file InventoryParserHelper.cpp.

References AIRINV::LegCabinStruct::\_bucketList, AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate, AIRINV::FlightDateStruct::\_itBucket, AIRINV::FlightDateStruct::\_itLegCabin, and AIRINV::BucketStruct::\_yieldRangeUpperValue.

### 10.204.4 Member Data Documentation

#### 10.204.4.1 FlightDateStruct& AIRINV::InventoryParserHelper::ParserSemanticAction::\_flightDate [inherited]

Actor Context.

Definition at line 33 of file InventoryParserHelper.hpp.

Referenced by AIRINV::InventoryParserHelper::doEndFlightDate::operator(), AIRINV::InventoryParserHelper::storeFClasses::operator(), AIRINV::InventoryParserHelper::storeFamilyCode::operator(), AIRINV::InventoryParserHelper::storeRevenueAvailability::operator(), AIRINV::InventoryParserHelper::storeSegmentAvailability::operator(), AIRINV::InventoryParserHelper::storeClassAvailability::operator(), AIRINV::InventoryParserHelper::storeClassETB::operator(), AIRINV::InventoryParserHelper::storeNbOfWLBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfStaffBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfGroupBkgs::operator(), AIRINV::InventoryParserHelper::storeNbOfBkgs::operator(), AIRINV::InventoryParserHelper::storeOverbooking::operator(), AIRINV::InventoryParserHelper::storeNoShow::operator(), AIRINV::InventoryParserHelper::storeNego::operator(), AIRINV::InventoryParserHelper::storeProtection::operator(), AIRINV::InventoryParserHelper::storeCumulatedProtection::operator(), AIRINV::InventoryParserHelper::storeParentSubclassCode::operator(), AIRINV::InventoryParserHelper::storeParentClassCode::operator(), AIRINV::InventoryParserHelper::storeSubclassCode::operator(), AIRINV::InventoryParserHelper::storeClassCode::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter::operator(), AIRINV::InventoryParserHelper::storeSegmentCabinCode::operator(), AIRINV::InventoryParserHelper::storeSegmentOffPoint::operator(), AIRINV::InventoryParserHelper::storeSegmentBoardingPoint::operator(), AIRINV::InventoryParserHelper::storeSeatIndex::operator(), AIRINV::InventoryParserHelper::store-

BucketAvaibility::operator(), operator(), AIRINV::InventoryParserHelper::storeETB::operator(),  
 AIRINV::InventoryParserHelper::storeACP::operator(), AIRINV::InventoryParserHelper::store-  
 GAV::operator(), AIRINV::InventoryParserHelper::storeNAV::operator(), AIRINV::InventoryParser-  
 Helper::storeBookingCounter::operator(), AIRINV::InventoryParserHelper::storeUPR::operator(),  
 AIRINV::InventoryParserHelper::storeAU::operator(), AIRINV::InventoryParserHelper::store-  
 SaleableCapacity::operator(), AIRINV::InventoryParserHelper::storeLegCabinCode::operator(),  
 AIRINV::InventoryParserHelper::storeOffTime::operator(), AIRINV::InventoryParserHelper::store-  
 OffDate::operator(), AIRINV::InventoryParserHelper::storeBoardingTime::operator(),  
 AIRINV::InventoryParserHelper::storeBoardingDate::operator(), AIRINV::InventoryParser-  
 Helper::storeOperatingFlightNumber::operator(), AIRINV::InventoryParserHelper::storeOperating-  
 AirlineCode::operator(), AIRINV::InventoryParserHelper::storeLegOffPoint::operator(),  
 AIRINV::InventoryParserHelper::storeLegBoardingPoint::operator(), AIRINV::InventoryParser-  
 Helper::storeFlightVisibilityCode::operator(), AIRINV::InventoryParserHelper::storeFlightType-  
 Code::operator(), AIRINV::InventoryParserHelper::storeFlightDate::operator(), AIRINV::Inventory-  
 ParserHelper::storeFlightNumber::operator(), AIRINV::InventoryParserHelper::storeAirline-  
 Code::operator(), and AIRINV::InventoryParserHelper::storeSnapshotDate::operator().

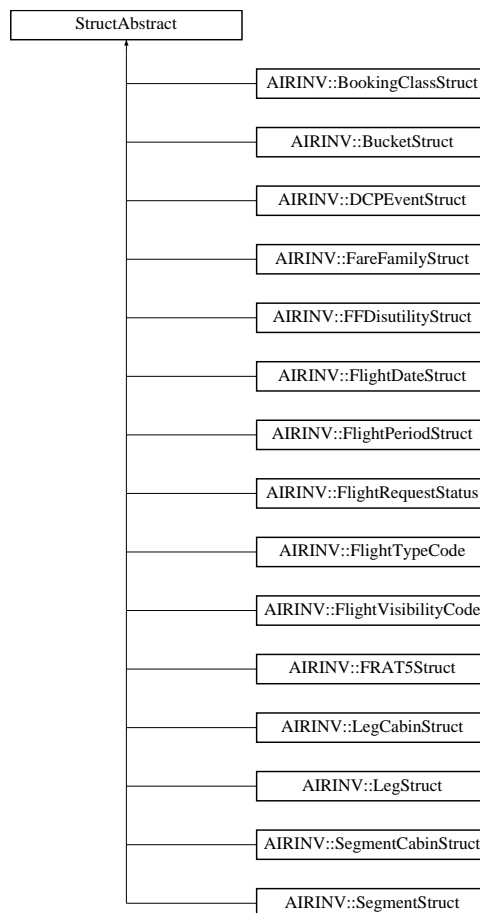
The documentation for this struct was generated from the following files:

- [airinv/command/InventoryParserHelper.hpp](#)
- [airinv/command/InventoryParserHelper.cpp](#)

## 10.205 StructAbstract Class Reference

Inheritance diagram for StructAbstract::





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

- [airinv/bom/SegmentStruct.hpp](#)
- [airinv/bom/FFDisutilityStruct.hpp](#)
- [airinv/bom/LegCabinStruct.hpp](#)
- [airinv/bom/LegStruct.hpp](#)
- [airinv/bom/DCPEventStruct.hpp](#)
- [airinv/FlightRequestStatus.hpp](#)
- [airinv/bom/FlightPeriodStruct.hpp](#)
- [airinv/bom/BucketStruct.hpp](#)
- [airinv/bom/FRAT5Struct.hpp](#)
- [airinv/basic/FlightTypeCode.hpp](#)
- [airinv/basic/FlightVisibilityCode.hpp](#)
- [airinv/bom/FlightDateStruct.hpp](#)
- [airinv/bom/FareFamilyStruct.hpp](#)
- [airinv/bom/BookingClassStruct.hpp](#)
- [airinv/bom/SegmentCabinStruct.hpp](#)

## 11 AirInv File Documentation

### 11.1 airinv/AIRINV\_Master\_Service.hpp File Reference

```
#include <string>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_file.hpp>
#include <stdair/stdair_service_types.hpp>
#include <stdair/stdair_inventory_types.hpp>
#include <stdair/stdair_maths_types.hpp>
#include <stdair/bom/BomIDTypes.hpp>
#include <airrac/AIRRAC_Types.hpp>
#include <sevmgr/SEVMGR_Types.hpp>
#include <airinv/AIRINV_Types.hpp>
```

#### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

#### Classes

- class [AIRINV::AIRINV\\_Master\\_Service](#)  
*Interface for the [AIRINV](#) Services.*

### 11.2 airinv/AIRINV\_Service.hpp File Reference

```
#include <string>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_service_types.hpp>
#include <stdair/basic/JSonCommand.hpp>
#include <stdair/bom/RMEventTypes.hpp>
#include <stdair/bom/BomIDTypes.hpp>
#include <airrac/AIRRAC_Types.hpp>
#include <sevmgr/SEVMGR_Types.hpp>
```

#### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

## Classes

- class [AIRINV::AIRINV\\_Service](#)  
*Interface for the [AIRINV](#) Services.*

## 11.3 airinv/AIRINV\_Types.hpp File Reference

```
#include <map>
#include <boost/shared_ptr.hpp>
#include <stdair/stdair_exceptions.hpp>
#include <stdair/stdair_file.hpp>
#include <stdair/stdair_inventory_types.hpp>
```

## Namespaces

- namespace [AIRINV](#)

## Classes

- class [AIRINV::InventoryFileParsingFailedException](#)
- class [AIRINV::ScheduleFileParsingFailedException](#)
- class [AIRINV::MissingPartnerFlightDateWithinScheduleFile](#)
- class [AIRINV::FRAT5FileParsingFailedException](#)
- class [AIRINV::FFDisutilityFileParsingFailedException](#)
- class [AIRINV::SegmentDateNotFoundException](#)
- class [AIRINV::InventoryInputFileNotFoundException](#)
- class [AIRINV::ScheduleInputFileNotFoundException](#)
- class [AIRINV::FRAT5InputFileNotFoundException](#)
- class [AIRINV::FFDisutilityInputFileNotFoundException](#)
- class [AIRINV::FlightDateDuplicationException](#)
- class [AIRINV::BookingException](#)
- class [AIRINV::InventoryNotFoundException](#)
- class [AIRINV::FlightDateNotFoundException](#)
- class [AIRINV::InventoryFilePath](#)

## Typedefs

- typedef boost::shared\_ptr< [AIRINV\\_Service](#) > [AIRINV::AIRINV\\_ServicePtr\\_T](#)
- typedef boost::shared\_ptr< [AIRINV\\_Master\\_Service](#) > [AIRINV::AIRINV\\_Master\\_ServicePtr\\_T](#)
- typedef std::map< const stdair::AirlineCode\_T, [AIRINV\\_ServicePtr\\_T](#) > [AIRINV::AIRINV\\_ServicePtr\\_Map\\_T](#)
- typedef std::map< const stdair::DTD\_T, double > [AIRINV::FRAT5Curve\\_T](#)

## 11.4 airinv/basic/BasConst.cpp File Reference

```
#include <airinv/basic/BasConst_General.hpp>
#include <airinv/basic/BasConst_Curves.hpp>
#include <airinv/basic/BasConst_AIRINV_Service.hpp>
```

### Namespaces

- namespace [AIRINV](#)

### Variables

- const std::string [AIRINV::DEFAULT\\_AIRLINE\\_CODE](#) = "BA"
- const [FRAT5Curve\\_T](#) [AIRINV::DEFAULT\\_PICKUP\\_FRAT5\\_CURVE](#)

## 11.5 airinv/basic/BasConst\_AIRINV\_Service.hpp File Reference

```
#include <string>
```

### Namespaces

- namespace [AIRINV](#)

### Variables

- const std::string [AIRINV::DEFAULT\\_AIRLINE\\_CODE](#)

## 11.6 airinv/basic/BasConst\_Curves.hpp File Reference

```
#include <airinv/AIRINV_Types.hpp>
```

### Namespaces

- namespace [AIRINV](#)

### Classes

- struct [AIRINV::DefaultMap](#)

### Variables

- const [FRAT5Curve\\_T](#) [AIRINV::DEFAULT\\_PICKUP\\_FRAT5\\_CURVE](#)

## 11.7 airinv/basic/BasConst\_General.hpp File Reference

### Namespaces

- namespace [AIRINV](#)

## 11.8 airinv/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/config.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 [AIRINV](#)

### Typedefs

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

## 11.9 airinv/basic/FlightRequestStatus.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/service/Logger.hpp>
#include <airinv/AIRINV_Types.hpp>
#include <airinv/FlightRequestStatus.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.10 airinv/basic/FlightTypeCode.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/service/Logger.hpp>
#include <airinv/AIRINV_Types.hpp>
#include <airinv/basic/FlightTypeCode.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.11 airinv/basic/FlightTypeCode.hpp File Reference

```
#include <string>
#include <stdair/basic/StructAbstract.hpp>
```

### Namespaces

- namespace [AIRINV](#)

### Classes

- struct [AIRINV::FlightTypeCode](#)

## 11.12 airinv/basic/FlightVisibilityCode.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/service/Logger.hpp>
#include <airinv/AIRINV_Types.hpp>
```

```
#include <airinv/basic/FlightVisibilityCode.hpp>
```

#### Namespaces

- namespace [AIRINV](#)

### 11.13 airinv/basic/FlightVisibilityCode.hpp File Reference

```
#include <string>
```

```
#include <stdair/basic/StructAbstract.hpp>
```

#### Namespaces

- namespace [AIRINV](#)

#### Classes

- struct [AIRINV::FlightVisibilityCode](#)

### 11.14 airinv/batches/airinv\_parseInventory.cpp File Reference

### 11.15 airinv/batches/parseInventory.cpp File Reference

### 11.16 airinv/bom/AirportList.hpp File Reference

```
#include <set>
```

```
#include <vector>
```

```
#include <stdair/stdair_basic_types.hpp>
```

#### Namespaces

- namespace [AIRINV](#)

#### Typedefs

- typedef std::set< stdair::AirportCode\_T > [AIRINV::AirportList\\_T](#)
- typedef std::vector< stdair::AirportCode\_T > [AIRINV::AirportOrderedList\\_T](#)

### 11.17 airinv/bom/BomAbstract.cpp File Reference

```
#include <airinv/bom/BomAbstract.hpp>
```

#### Namespaces

- namespace [AIRINV](#)

## 11.18 airinv/bom/BomAbstract.hpp File Reference

```
#include <iosfwd>
#include <string>
```

### Namespaces

- namespace [AIRINV](#)

### Classes

- class [AIRINV::BomAbstract](#)

### Functions

- `template<class charT, class traits> std::basic_ostream< charT, traits > & operator<< (std::basic_ostream< charT, traits > &ioOut, const AIRINV::BomAbstract &iBom)`
- `template<class charT, class traits> std::basic_istream< charT, traits > & operator>> (std::basic_istream< charT, traits > &ioIn, AIRINV::BomAbstract &iBom)`

#### 11.18.1 Function Documentation

**11.18.1.1** `template<class charT, class traits> std::basic_ostream<charT, traits>& operator<< (std::basic_ostream< charT, traits > &ioOut, const AIRINV::BomAbstract &iBom)` `[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 56 of file BomAbstract.hpp.

**11.18.1.2** `template<class charT, class traits> std::basic_istream<charT, traits>& operator>> (std::basic_istream< charT, traits > &ioIn, AIRINV::BomAbstract &iBom)` `[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 84 of file BomAbstract.hpp.

References [AIRINV::BomAbstract::fromStream\(\)](#).

## 11.19 airinv/bom/BomRootHelper.cpp File Reference

```
#include <cassert>
#include <stdair/bom/BomManager.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/bom/Inventory.hpp>
#include <airinv/bom/BomRootHelper.hpp>
#include <airinv/bom/InventoryHelper.hpp>
```



### Namespaces

- namespace [AIRINV](#)

## 11.20 airinv/bom/BomRootHelper.hpp File Reference

### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

### Classes

- class [AIRINV::BomRootHelper](#)

## 11.21 airinv/bom/BookingClassHelper.cpp File Reference

```
#include <cassert>
#include <stdair/bom/BookingClass.hpp>
#include <airinv/bom/BookingClassHelper.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.22 airinv/bom/BookingClassHelper.hpp File Reference

### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

### Classes

- class [AIRINV::BookingClassHelper](#)

## 11.23 airinv/bom/BookingClassStruct.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/basic/BasConst_General.hpp>
#include <stdair/bom/BookingClass.hpp>
#include <airinv/bom/BookingClassStruct.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.24 airinv/bom/BookingClassStruct.hpp File Reference

```
#include <string>
#include <vector>
#include <stdair/stdair_inventory_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
#include <airinv/AIRINV_Types.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

### Classes

- struct [AIRINV::BookingClassStruct](#)

### Typedefs

- typedef std::vector< BookingClassStruct > [AIRINV::BookingClassStructList\\_T](#)

## 11.25 airinv/bom/BucketStruct.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/basic/BasConst_General.hpp>
#include <stdair/bom/Bucket.hpp>
#include <airinv/bom/BucketStruct.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.26 airinv/bom/BucketStruct.hpp File Reference

```
#include <string>
#include <vector>
#include <stdair/stdair_inventory_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
#include <airinv/AIRINV_Types.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

### Classes

- struct [AIRINV::BucketStruct](#)  
*Utility Structure for the parsing of Bucket structures.*

### Typedefs

- typedef std::vector< BucketStruct > [AIRINV::BucketStructList\\_T](#)

## 11.27 airinv/bom/DCPEventStruct.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <vector>
#include <stdair/basic/BasConst_General.hpp>
#include <stdair/service/Logger.hpp>
#include <airinv/AIRINV_Types.hpp>
#include <airinv/bom/DCPEventStruct.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.28 airinv/bom/DCPEventStruct.hpp File Reference

```
#include <string>
#include <vector>
#include <stdair/stdair_demand_types.hpp>
#include <stdair/stdair_inventory_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
#include <stdair/basic/BasParserTypes.hpp>
#include <airinv/AIRINV_Types.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## Classes

- struct [AIRINV::DCPEventStruct](#)

## 11.29 airinv/bom/FareFamilyStruct.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/basic/BasConst_Inventory.hpp>
#include <stdair/bom/FareFamily.hpp>
#include <airinv/bom/FareFamilyStruct.hpp>
```

## Namespaces

- namespace [AIRINV](#)

## 11.30 airinv/bom/FareFamilyStruct.hpp File Reference

```
#include <string>
#include <vector>
#include <stdair/stdair_inventory_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
#include <airinv/bom/BookingClassStruct.hpp>
```

## Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

## Classes

- struct [AIRINV::FareFamilyStruct](#)  
*Utility Structure for the parsing of fare family details.*

## Typedefs

- typedef std::vector< FareFamilyStruct > [AIRINV::FareFamilyStructList\\_T](#)

## 11.31 airinv/bom/FFDisutilityStruct.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/service/Logger.hpp>
```

```
#include <airinv/bom/FFDisutilityStruct.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.32 airinv/bom/FFDisutilityStruct.hpp File Reference

```
#include <string>
```

```
#include <stdair/stdair_rm_types.hpp>
```

### Namespaces

- namespace [AIRINV](#)

### Classes

- struct [AIRINV::FFDisutilityStruct](#)

## 11.33 airinv/bom/FlightDateHelper.cpp File Reference

```
#include <cassert>
```

```
#include <stdair/basic/BasConst_Inventory.hpp>
```

```
#include <stdair/bom/BomManager.hpp>
```

```
#include <stdair/bom/FlightDate.hpp>
```

```
#include <stdair/bom/SegmentDate.hpp>
```

```
#include <stdair/bom/SegmentCabin.hpp>
```

```
#include <stdair/bom/LegCabin.hpp>
```

```
#include <airinv/bom/FlightDateHelper.hpp>
```

```
#include <airinv/bom/SegmentDateHelper.hpp>
```

```
#include <airinv/bom/SegmentCabinHelper.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.34 airinv/bom/FlightDateHelper.hpp File Reference

```
#include <stdair/stdair_basic_types.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

## Classes

- class [AIRINV::FlightDateHelper](#)

## 11.35 airinv/bom/FlightDateStruct.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/basic/BasConst_General.hpp>
#include <stdair/service/Logger.hpp>
#include <airinv/AIRINV_Types.hpp>
#include <airinv/bom/FlightDateStruct.hpp>
```

## Namespaces

- namespace [AIRINV](#)

## 11.36 airinv/bom/FlightDateStruct.hpp File Reference

```
#include <string>
#include <stdair/stdair_inventory_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
#include <stdair/bom/DoWStruct.hpp>
#include <airinv/basic/FlightTypeCode.hpp>
#include <airinv/basic/FlightVisibilityCode.hpp>
#include <airinv/bom/LegStruct.hpp>
#include <airinv/bom/LegCabinStruct.hpp>
#include <airinv/bom/BucketStruct.hpp>
#include <airinv/bom/SegmentStruct.hpp>
#include <airinv/bom/SegmentCabinStruct.hpp>
#include <airinv/bom/FareFamilyStruct.hpp>
#include <airinv/bom/AirportList.hpp>
```

## Namespaces

- namespace [AIRINV](#)

## Classes

- struct [AIRINV::FlightDateStruct](#)

## 11.37 airinv/bom/FlightPeriodStruct.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/basic/BasConst_Period_BOM.hpp>
#include <stdair/service/Logger.hpp>
#include <airinv/AIRINV_Types.hpp>
#include <airinv/bom/FlightPeriodStruct.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.38 airinv/bom/FlightPeriodStruct.hpp File Reference

```
#include <string>
#include <stdair/stdair_inventory_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
#include <stdair/bom/DoWStruct.hpp>
#include <airinv/bom/LegCabinStruct.hpp>
#include <airinv/bom/LegStruct.hpp>
#include <airinv/bom/SegmentStruct.hpp>
#include <airinv/bom/SegmentCabinStruct.hpp>
#include <airinv/bom/FareFamilyStruct.hpp>
#include <airinv/bom/AirportList.hpp>
```

### Namespaces

- namespace [AIRINV](#)

### Classes

- struct [AIRINV::FlightPeriodStruct](#)

## 11.39 airinv/bom/FRAT5Struct.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/service/Logger.hpp>
#include <airinv/bom/FRAT5Struct.hpp>
```

## Namespaces

- namespace [AIRINV](#)

## 11.40 airinv/bom/FRAT5Struct.hpp File Reference

```
#include <string>
#include <stdair/stdair_rm_types.hpp>
```

## Namespaces

- namespace [AIRINV](#)

## Classes

- struct [AIRINV::FRAT5Struct](#)

## 11.41 airinv/bom/InventoryHelper.cpp File Reference

```
#include <cassert>
#include <stdair/bom/BomRetriever.hpp>
#include <stdair/bom/BomManager.hpp>
#include <stdair/bom/Inventory.hpp>
#include <stdair/bom/FlightDate.hpp>
#include <stdair/bom/SegmentDate.hpp>
#include <stdair/bom/SegmentCabin.hpp>
#include <stdair/bom/FareFamily.hpp>
#include <stdair/bom/BookingClass.hpp>
#include <stdair/bom/SegmentSnapshotTable.hpp>
#include <stdair/bom/TravelSolutionStruct.hpp>
#include <stdair/service/Logger.hpp>
#include <stdair/bom/LegCabin.hpp>
#include <airinv/bom/InventoryHelper.hpp>
#include <airinv/bom/FlightDateHelper.hpp>
#include <airinv/bom/SegmentSnapshotTableHelper.hpp>
#include <airinv/bom/SegmentCabinHelper.hpp>
```

## Namespaces

- namespace [AIRINV](#)



## 11.42 airinv/bom/InventoryHelper.hpp File Reference

```
#include <string>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/bom/BomIDTypes.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

### Classes

- class [AIRINV::InventoryHelper](#)

## 11.43 airinv/bom/LegCabinHelper.cpp File Reference

```
#include <cassert>
#include <stdair/bom/LegCabin.hpp>
#include <airinv/bom/LegCabinHelper.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.44 airinv/bom/LegCabinHelper.hpp File Reference

### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

### Classes

- class [AIRINV::LegCabinHelper](#)

## 11.45 airinv/bom/LegCabinStruct.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/bom/LegCabin.hpp>
#include <airinv/bom/LegCabinStruct.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.46 airinv/bom/LegCabinStruct.hpp File Reference

```
#include <string>
#include <vector>
#include <stdair/stdair_inventory_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
#include <airinv/bom/BucketStruct.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

### Classes

- struct [AIRINV::LegCabinStruct](#)

### Typedefs

- typedef std::vector< LegCabinStruct > [AIRINV::LegCabinStructList\\_T](#)

## 11.47 airinv/bom/LegStruct.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/basic/BasConst_General.hpp>
#include <stdair/bom/LegDate.hpp>
#include <airinv/bom/LegStruct.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.48 airinv/bom/LegStruct.hpp File Reference

```
#include <string>
#include <vector>
#include <stdair/stdair_inventory_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
#include <airinv/bom/LegCabinStruct.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

### Classes

- struct [AIRINV::LegStruct](#)

### Typedefs

- typedef std::vector< LegStruct > [AIRINV::LegStructList\\_T](#)

## 11.49 airinv/bom/SegmentCabinHelper.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <limits>
#include <stdair/basic/BasConst_Inventory.hpp>
#include <stdair/basic/float_utils.hpp>
#include <stdair/bom/BomManager.hpp>
#include <stdair/bom/FlightDate.hpp>
#include <stdair/bom/LegCabin.hpp>
#include <stdair/bom/SegmentCabin.hpp>
#include <stdair/bom/FareFamily.hpp>
#include <stdair/bom/BookingClass.hpp>
#include <stdair/bom/SimpleNestingStructure.hpp>
#include <stdair/bom/NestingNode.hpp>
#include <stdair/bom/Policy.hpp>
#include <stdair/factory/FacBomManager.hpp>
#include <airinv/bom/SegmentCabinHelper.hpp>
#include <airinv/bom/FlightDateHelper.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.50 airinv/bom/SegmentCabinHelper.hpp File Reference

```
#include <stdair/stdair_basic_types.hpp>
#include <stdair/bom/FareFamilyTypes.hpp>
```

## Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

## Classes

- class [AIRINV::SegmentCabinHelper](#)  
*Class representing the actual business functions for an airline segment-cabin.*

## 11.51 airinv/bom/SegmentCabinStruct.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/bom/SegmentCabin.hpp>
#include <airinv/bom/SegmentCabinStruct.hpp>
```

## Namespaces

- namespace [AIRINV](#)

## 11.52 airinv/bom/SegmentCabinStruct.hpp File Reference

```
#include <string>
#include <vector>
#include <stdair/stdair_inventory_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
#include <airinv/bom/FareFamilyStruct.hpp>
```

## Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

## Classes

- struct [AIRINV::SegmentCabinStruct](#)  
*Utility Structure for the parsing of SegmentCabin details.*

## Typedefs

- typedef std::vector< SegmentCabinStruct > [AIRINV::SegmentCabinStructList\\_T](#)

## 11.53 airinv/bom/SegmentDateHelper.cpp File Reference

```
#include <cassert>
#include <stdair/basic/BasConst_General.hpp>
#include <stdair/bom/BomManager.hpp>
#include <stdair/bom/SegmentDate.hpp>
#include <stdair/bom/SegmentCabin.hpp>
#include <stdair/bom/LegDate.hpp>
#include <airinv/bom/SegmentDateHelper.hpp>
#include <airinv/bom/SegmentCabinHelper.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.54 airinv/bom/SegmentDateHelper.hpp File Reference

### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

### Classes

- class [AIRINV::SegmentDateHelper](#)

## 11.55 airinv/bom/SegmentSnapshotTableHelper.cpp File Reference

```
#include <cassert>
#include <cmath>
#include <stdair/basic/BasConst_Inventory.hpp>
#include <stdair/bom/BomRetriever.hpp>
#include <stdair/bom/BomManager.hpp>
#include <stdair/bom/SegmentDate.hpp>
#include <stdair/bom/SegmentCabin.hpp>
#include <stdair/bom/FareFamily.hpp>
#include <stdair/bom/BookingClass.hpp>
#include <stdair/bom/SegmentSnapshotTable.hpp>
#include <stdair/service/Logger.hpp>
#include <airinv/basic/BasConst_Curves.hpp>
#include <airinv/bom/SegmentSnapshotTableHelper.hpp>
```

```
#include <airinv/bom/FlightDateHelper.hpp>
#include <airinv/bom/SegmentCabinHelper.hpp>
```

#### Namespaces

- namespace [AIRINV](#)

### 11.56 `airinv/bom/SegmentSnapshotTableHelper.hpp` File Reference

```
#include <string>
#include <stdair/stdair_basic_types.hpp>
```

#### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

#### Classes

- class [AIRINV::SegmentSnapshotTableHelper](#)

### 11.57 `airinv/bom/SegmentStruct.cpp` File Reference

```
#include <cassert>
#include <stdair/bom/SegmentDate.hpp>
#include <airinv/bom/SegmentStruct.hpp>
```

#### Namespaces

- namespace [AIRINV](#)

### 11.58 `airinv/bom/SegmentStruct.hpp` File Reference

```
#include <string>
#include <vector>
#include <stdair/stdair_inventory_types.hpp>
#include <stdair/basic/StructAbstract.hpp>
#include <airinv/bom/SegmentCabinStruct.hpp>
```

#### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

## Classes

- struct [AIRINV::SegmentStruct](#)

## Typedefs

- typedef std::vector< SegmentStruct > [AIRINV::SegmentStructList\\_T](#)

## 11.59 airinv/command/FFDisutilityParser.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/service/Logger.hpp>
#include <airinv/command/FFDisutilityParserHelper.hpp>
#include <airinv/command/FFDisutilityParser.hpp>
#include <airinv/command/InventoryManager.hpp>
```

## Namespaces

- namespace [AIRINV](#)

## 11.60 airinv/command/FFDisutilityParser.hpp File Reference

```
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_file.hpp>
#include <stdair/command/CmdAbstract.hpp>
```

## Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

## Classes

- class [AIRINV::FFDisutilityParser](#)  
*Class wrapping the parser entry point.*

## 11.61 airinv/command/FFDisutilityParserHelper.cpp File Reference

```
#include <cassert>
#include <ostream>
```

```
#include <stdair/stdair_exceptions.hpp>
#include <stdair/stdair_types.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/service/Logger.hpp>
#include <airinv/command/FFDisutilityParserHelper.hpp>
```

### Namespaces

- namespace [AIRINV](#)
- namespace [AIRINV::FFDisutilityParserHelper](#)

### Functions

- [repeat\\_p\\_t AIRINV::FFDisutilityParserHelper::key\\_p](#) ([chset\\_t](#)("0-9A-Z").derived(), 1, 10)

## 11.62 airinv/command/FFDisutilityParserHelper.hpp File Reference

```
#include <string>
#include <stdair/command/CmdAbstract.hpp>
#include <airinv/AIRINV_Types.hpp>
#include <airinv/basic/BasParserTypes.hpp>
#include <airinv/bom/FFDisutilityStruct.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)
- namespace [AIRINV::FFDisutilityParserHelper](#)

### Classes

- struct [AIRINV::FFDisutilityParserHelper::ParserSemanticAction](#)
- struct [AIRINV::FFDisutilityParserHelper::storeCurveKey](#)
- struct [AIRINV::FFDisutilityParserHelper::storeDTD](#)
- struct [AIRINV::FFDisutilityParserHelper::storeFFDisutilityValue](#)
- struct [AIRINV::FFDisutilityParserHelper::doEndCurve](#)
- struct [AIRINV::FFDisutilityParserHelper::FFDisutilityParser](#)
- struct [AIRINV::FFDisutilityParserHelper::FFDisutilityParser::definition< ScannerT >](#)
- class [AIRINV::FFDisutilityFileParser](#)

## 11.63 airinv/command/FRAT5Parser.cpp File Reference

```
#include <cassert>
#include <sstream>
```



```
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/service/Logger.hpp>
#include <airinv/command/FRAT5ParserHelper.hpp>
#include <airinv/command/FRAT5Parser.hpp>
#include <airinv/command/InventoryManager.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.64 airinv/command/FRAT5Parser.hpp File Reference

```
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_file.hpp>
#include <stdair/command/CmdAbstract.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

### Classes

- class [AIRINV::FRAT5Parser](#)  
*Class wrapping the parser entry point.*

## 11.65 airinv/command/FRAT5ParserHelper.cpp File Reference

```
#include <cassert>
#include <ostream>
#include <stdair/stdair_exceptions.hpp>
#include <stdair/stdair_types.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/service/Logger.hpp>
#include <airinv/command/FRAT5ParserHelper.hpp>
```

### Namespaces

- namespace [AIRINV](#)
- namespace [AIRINV::FRAT5ParserHelper](#)

## Functions

- [repeat\\_p\\_t](#) [AIRINV::FRAT5ParserHelper::key\\_p](#) ([chset\\_t](#)("0-9A-Z").derived(), 1, 10)

## 11.66 airinv/command/FRAT5ParserHelper.hpp File Reference

```
#include <string>
#include <stdair/command/CmdAbstract.hpp>
#include <airinv/AIRINV_Types.hpp>
#include <airinv/basic/BasParserTypes.hpp>
#include <airinv/bom/FRAT5Struct.hpp>
```

## Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)
- namespace [AIRINV::FRAT5ParserHelper](#)

## Classes

- struct [AIRINV::FRAT5ParserHelper::ParserSemanticAction](#)
- struct [AIRINV::FRAT5ParserHelper::storeCurveKey](#)
- struct [AIRINV::FRAT5ParserHelper::storeDTD](#)
- struct [AIRINV::FRAT5ParserHelper::storeFRAT5Value](#)
- struct [AIRINV::FRAT5ParserHelper::doEndCurve](#)
- struct [AIRINV::FRAT5ParserHelper::FRAT5Parser](#)
- struct [AIRINV::FRAT5ParserHelper::FRAT5Parser::definition< ScannerT >](#)
- class [AIRINV::FRAT5FileParser](#)

## 11.67 airinv/command/InventoryBuilder.cpp File Reference

```
#include <cassert>
#include <boost/date_time/date_iterator.hpp>
#include <stdair/basic/BasConst_BookingClass.hpp>
#include <stdair/basic/BasConst_Yield.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/FlightDate.hpp>
#include <stdair/bom/SegmentDate.hpp>
#include <stdair/bom/SegmentCabin.hpp>
```

```
#include <stdair/bom/FareFamily.hpp>
#include <stdair/bom/BookingClass.hpp>
#include <stdair/bom/LegDate.hpp>
#include <stdair/bom/LegCabin.hpp>
#include <stdair/bom/Bucket.hpp>
#include <stdair/bom/BomKeyManager.hpp>
#include <stdair/bom/ParsedKey.hpp>
#include <stdair/bom/BomRetriever.hpp>
#include <stdair/command/CmdCloneBomManager.hpp>
#include <stdair/factory/FacBom.hpp>
#include <stdair/factory/FacBomManager.hpp>
#include <stdair/service/Logger.hpp>
#include <airinv/AIRINV_Types.hpp>
#include <airinv/bom/FlightDateStruct.hpp>
#include <airinv/command/InventoryBuilder.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.68 airinv/command/InventoryBuilder.hpp File Reference

```
#include <stdair/command/CmdAbstract.hpp>
#include <airinv/AIRINV_Types.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)
- namespace [AIRINV::InventoryParserHelper](#)

### Classes

- class [AIRINV::InventoryBuilder](#)  
*Class handling the generation / instantiation of the Inventory BOM.*

## 11.69 airinv/command/InventoryGenerator.cpp File Reference

```
#include <cassert>
#include <boost/date_time/date_iterator.hpp>
#include <stdair/stdair_types.hpp>
```

```
#include <stdair/basic/BasConst_Inventory.hpp>
#include <stdair/basic/BasConst_SellUpCurves.hpp>
#include <stdair/bom/BomManager.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/bom/Inventory.hpp>
#include <stdair/bom/AirlineFeature.hpp>
#include <stdair/bom/FlightDate.hpp>
#include <stdair/bom/SegmentDate.hpp>
#include <stdair/bom/SegmentCabin.hpp>
#include <stdair/bom/FareFamily.hpp>
#include <stdair/bom/BookingClass.hpp>
#include <stdair/bom/LegDate.hpp>
#include <stdair/bom/LegCabin.hpp>
#include <stdair/bom/SimpleNestingStructure.hpp>
#include <stdair/bom/NestingNode.hpp>
#include <stdair/bom/Policy.hpp>
#include <stdair/bom/Bucket.hpp>
#include <stdair/bom/BomKeyManager.hpp>
#include <stdair/factory/FacBomManager.hpp>
#include <stdair/service/Logger.hpp>
#include <airinv/bom/FlightPeriodStruct.hpp>
#include <airinv/command/InventoryGenerator.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.70 `airinv/command/InventoryGenerator.hpp` File Reference

```
#include <stdair/command/CmdAbstract.hpp>
#include <airinv/AIRINV_Types.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)
- namespace [AIRINV::ScheduleParserHelper](#)

### Classes

- class [AIRINV::InventoryGenerator](#)

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

## 11.71 airinv/command/InventoryManager.cpp File Reference

```
#include <exception>
#include <algorithm>
#include <boost/make_shared.hpp>
#include <stdair/basic/BasConst_Inventory.hpp>
#include <stdair/basic/BasConst_BomDisplay.hpp>
#include <stdair/bom/BomManager.hpp>
#include <stdair/bom/BomKeyManager.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/bom/Inventory.hpp>
#include <stdair/bom/FlightDate.hpp>
#include <stdair/bom/SegmentDate.hpp>
#include <stdair/bom/SegmentCabin.hpp>
#include <stdair/bom/LegDate.hpp>
#include <stdair/bom/LegCabin.hpp>
#include <stdair/bom/FareFamily.hpp>
#include <stdair/bom/BookingClass.hpp>
#include <stdair/bom/SegmentSnapshotTable.hpp>
#include <stdair/bom/TravelSolutionStruct.hpp>
#include <stdair/bom/FareOptionStruct.hpp>
#include <stdair/bom/EventStruct.hpp>
#include <stdair/bom/SnapshotStruct.hpp>
#include <stdair/bom/RMEventStruct.hpp>
#include <stdair/bom/BomRetriever.hpp>
#include <stdair/factory/FacBomManager.hpp>
#include <stdair/factory/FacBom.hpp>
#include <stdair/service/Logger.hpp>
#include <sevmgr/SEVMGR_Service.hpp>
#include <airinv/AIRINV_Types.hpp>
#include <airinv/bom/BomRootHelper.hpp>
#include <airinv/bom/InventoryHelper.hpp>
#include <airinv/bom/FlightDateHelper.hpp>
#include <airinv/bom/SegmentCabinHelper.hpp>
#include <airinv/command/InventoryManager.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.72 `airinv/command/InventoryManager.hpp` File Reference

```
#include <string>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/bom/RMEventTypes.hpp>
#include <stdair/bom/BomIDTypes.hpp>
#include <sevmgr/SEVMGR_Types.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

### Classes

- class [AIRINV::InventoryManager](#)

### Typedefs

- typedef `std::map< const stdair::Date_T, stdair::SegmentCabin * >` [AIRINV::DepartureDateSegmentCabinMap\\_T](#)
- typedef `std::map< const std::string, DepartureDateSegmentCabinMap_T >` [AIRINV::SimilarSegmentCabinSetMap\\_T](#)

## 11.73 `airinv/command/InventoryParser.cpp` File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/service/Logger.hpp>
#include <airinv/command/InventoryParserHelper.hpp>
#include <airinv/command/InventoryParser.hpp>
#include <airinv/command/InventoryManager.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.74 airinv/command/InventoryParser.hpp File Reference

```
#include <stdair/stdair_basic_types.hpp>
#include <stdair/command/CmdAbstract.hpp>
#include <airinv/AIRINV_Types.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

### Classes

- class [AIRINV::InventoryParser](#)  
*Class wrapping the parser entry point.*

## 11.75 airinv/command/InventoryParserHelper.cpp File Reference

```
#include <cassert>
#include <stdair/service/Logger.hpp>
#include <stdair/stdair_exceptions.hpp>
#include <airinv/command/InventoryBuilder.hpp>
#include <airinv/command/InventoryParserHelper.hpp>
```

### Namespaces

- namespace [AIRINV](#)
- namespace [AIRINV::InventoryParserHelper](#)

### Functions

- [repeat\\_p\\_t AIRINV::InventoryParserHelper::airline\\_code\\_p](#) ([chset\\_t](#)("0-9A-Z").derived(), 2, 3)
- [bounded1\\_4\\_p\\_t AIRINV::InventoryParserHelper::flight\\_number\\_p](#) ([uint1\\_4\\_p](#).derived(), 0u, 9999u)
- [bounded2\\_p\\_t AIRINV::InventoryParserHelper::year\\_p](#) ([uint2\\_p](#).derived(), 0u, 99u)
- [bounded2\\_p\\_t AIRINV::InventoryParserHelper::month\\_p](#) ([uint2\\_p](#).derived(), 1u, 12u)
- [bounded2\\_p\\_t AIRINV::InventoryParserHelper::day\\_p](#) ([uint2\\_p](#).derived(), 1u, 31u)
- [repeat\\_p\\_t AIRINV::InventoryParserHelper::dow\\_p](#) ([chset\\_t](#)("0-1").derived().derived(), 7, 7)
- [repeat\\_p\\_t AIRINV::InventoryParserHelper::airport\\_p](#) ([chset\\_t](#)("0-9A-Z").derived(), 3, 3)
- [bounded1\\_2\\_p\\_t AIRINV::InventoryParserHelper::hours\\_p](#) ([uint1\\_2\\_p](#).derived(), 0u, 24u)
- [bounded2\\_p\\_t AIRINV::InventoryParserHelper::minutes\\_p](#) ([uint2\\_p](#).derived(), 0u, 59u)
- [bounded2\\_p\\_t AIRINV::InventoryParserHelper::seconds\\_p](#) ([uint2\\_p](#).derived(), 0u, 59u)
- [chset\\_t AIRINV::InventoryParserHelper::cabin\\_code\\_p](#) ("A-Z")
- [chset\\_t AIRINV::InventoryParserHelper::class\\_code\\_p](#) ("A-Z")
- [chset\\_t AIRINV::InventoryParserHelper::passenger\\_type\\_p](#) ("A-Z")
- [repeat\\_p\\_t AIRINV::InventoryParserHelper::class\\_code\\_list\\_p](#) ([chset\\_t](#)("A-Z").derived(), 1, 26)
- [bounded1\\_3\\_p\\_t AIRINV::InventoryParserHelper::stay\\_duration\\_p](#) ([uint1\\_3\\_p](#).derived(), 0u, 999u)

## Variables

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

## 11.76 airinv/command/InventoryParserHelper.hpp File Reference

```
#include <string>
#include <stdair/command/CmdAbstract.hpp>
#include <airinv/AIRINV_Types.hpp>
#include <airinv/basic/BasParserTypes.hpp>
#include <airinv/bom/FlightDateStruct.hpp>
```

## Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)
- namespace [AIRINV::InventoryParserHelper](#)

## Classes

- struct [AIRINV::InventoryParserHelper::ParserSemanticAction](#)
- struct [AIRINV::InventoryParserHelper::storeSnapshotDate](#)
- struct [AIRINV::InventoryParserHelper::storeAirlineCode](#)
- struct [AIRINV::InventoryParserHelper::storeFlightNumber](#)
- struct [AIRINV::InventoryParserHelper::storeFlightDate](#)
- struct [AIRINV::InventoryParserHelper::storeFlightTypeCode](#)
- struct [AIRINV::InventoryParserHelper::storeFlightVisibilityCode](#)
- struct [AIRINV::InventoryParserHelper::storeLegBoardingPoint](#)
- struct [AIRINV::InventoryParserHelper::storeLegOffPoint](#)
- struct [AIRINV::InventoryParserHelper::storeOperatingAirlineCode](#)
- struct [AIRINV::InventoryParserHelper::storeOperatingFlightNumber](#)
- struct [AIRINV::InventoryParserHelper::storeBoardingDate](#)
- struct [AIRINV::InventoryParserHelper::storeBoardingTime](#)
- struct [AIRINV::InventoryParserHelper::storeOffDate](#)
- struct [AIRINV::InventoryParserHelper::storeOffTime](#)
- struct [AIRINV::InventoryParserHelper::storeLegCabinCode](#)
- struct [AIRINV::InventoryParserHelper::storeSaleableCapacity](#)
- struct [AIRINV::InventoryParserHelper::storeAU](#)
- struct [AIRINV::InventoryParserHelper::storeUPR](#)
- struct [AIRINV::InventoryParserHelper::storeBookingCounter](#)
- struct [AIRINV::InventoryParserHelper::storeNAV](#)
- struct [AIRINV::InventoryParserHelper::storeGAV](#)



- struct [AIRINV::InventoryParserHelper::storeACP](#)
- struct [AIRINV::InventoryParserHelper::storeETB](#)
- struct [AIRINV::InventoryParserHelper::storeYieldUpperRange](#)
- struct [AIRINV::InventoryParserHelper::storeBucketAvailability](#)
- struct [AIRINV::InventoryParserHelper::storeSeatIndex](#)
- struct [AIRINV::InventoryParserHelper::storeSegmentBoardingPoint](#)
- struct [AIRINV::InventoryParserHelper::storeSegmentOffPoint](#)
- struct [AIRINV::InventoryParserHelper::storeSegmentCabinCode](#)
- struct [AIRINV::InventoryParserHelper::storeSegmentCabinBookingCounter](#)
- struct [AIRINV::InventoryParserHelper::storeClassCode](#)
- struct [AIRINV::InventoryParserHelper::storeSubclassCode](#)
- struct [AIRINV::InventoryParserHelper::storeParentClassCode](#)
- struct [AIRINV::InventoryParserHelper::storeParentSubclassCode](#)
- struct [AIRINV::InventoryParserHelper::storeCumulatedProtection](#)
- struct [AIRINV::InventoryParserHelper::storeProtection](#)
- struct [AIRINV::InventoryParserHelper::storeNego](#)
- struct [AIRINV::InventoryParserHelper::storeNoShow](#)
- struct [AIRINV::InventoryParserHelper::storeOverbooking](#)
- struct [AIRINV::InventoryParserHelper::storeNbOfBkgs](#)
- struct [AIRINV::InventoryParserHelper::storeNbOfGroupBkgs](#)
- struct [AIRINV::InventoryParserHelper::storeNbOfPendingGroupBkgs](#)
- struct [AIRINV::InventoryParserHelper::storeNbOfStaffBkgs](#)
- struct [AIRINV::InventoryParserHelper::storeNbOfWLBkgs](#)
- struct [AIRINV::InventoryParserHelper::storeClassETB](#)
- struct [AIRINV::InventoryParserHelper::storeClassAvailability](#)
- struct [AIRINV::InventoryParserHelper::storeSegmentAvailability](#)
- struct [AIRINV::InventoryParserHelper::storeRevenueAvailability](#)
- struct [AIRINV::InventoryParserHelper::storeFamilyCode](#)
- struct [AIRINV::InventoryParserHelper::storeFCClasses](#)
- struct [AIRINV::InventoryParserHelper::doEndFlightDate](#)
- struct [AIRINV::InventoryParserHelper::InventoryParser](#)
- struct [AIRINV::InventoryParserHelper::InventoryParser::definition< ScannerT >](#)
- class [AIRINV::InventoryFileParser](#)

## 11.77 airinv/command/ScheduleParser.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/service/Logger.hpp>
#include <airinv/command/ScheduleParserHelper.hpp>
#include <airinv/command/ScheduleParser.hpp>
#include <airinv/command/InventoryManager.hpp>
```

## Namespaces

- namespace [AIRINV](#)

## 11.78 airinv/command/ScheduleParser.hpp File Reference

```
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_file.hpp>
#include <stdair/command/CmdAbstract.hpp>
```

## Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

## Classes

- class [AIRINV::ScheduleParser](#)  
*Class wrapping the parser entry point.*

## 11.79 airinv/command/ScheduleParserHelper.cpp File Reference

```
#include <cassert>
#include <stdair/stdair_exceptions.hpp>
#include <stdair/stdair_types.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/service/Logger.hpp>
#include <airinv/command/InventoryGenerator.hpp>
#include <airinv/command/ScheduleParserHelper.hpp>
```

## Namespaces

- namespace [AIRINV](#)
- namespace [AIRINV::ScheduleParserHelper](#)

## Functions

- [repeat\\_p\\_t AIRINV::ScheduleParserHelper::airline\\_code\\_p](#) ([chset\\_t](#)("0-9A-Z").derived(), 2, 3)
- [bounded1\\_4\\_p\\_t AIRINV::ScheduleParserHelper::flight\\_number\\_p](#) ([uint1\\_4\\_p](#).derived(), 0u, 9999u)
- [bounded4\\_p\\_t AIRINV::ScheduleParserHelper::year\\_p](#) ([uint4\\_p](#).derived(), 2000u, 2099u)
- [bounded2\\_p\\_t AIRINV::ScheduleParserHelper::month\\_p](#) ([uint2\\_p](#).derived(), 1u, 12u)
- [bounded2\\_p\\_t AIRINV::ScheduleParserHelper::day\\_p](#) ([uint2\\_p](#).derived(), 1u, 31u)
- [repeat\\_p\\_t AIRINV::ScheduleParserHelper::dow\\_p](#) ([chset\\_t](#)("0-1").derived().derived(), 7, 7)

- `repeat_p_t AIRINV::ScheduleParserHelper::airport_p` (`chset_t("0-9A-Z").derived()`, 3, 3)
- `bounded2_p_t AIRINV::ScheduleParserHelper::hours_p` (`uint2_p.derived()`, 0u, 23u)
- `bounded2_p_t AIRINV::ScheduleParserHelper::minutes_p` (`uint2_p.derived()`, 0u, 59u)
- `bounded2_p_t AIRINV::ScheduleParserHelper::seconds_p` (`uint2_p.derived()`, 0u, 59u)
- `chset_t AIRINV::ScheduleParserHelper::cabin_code_p` ("A-Z")
- `repeat_p_t AIRINV::ScheduleParserHelper::key_p` (`chset_t("0-9A-Z").derived()`, 1, 10)
- `repeat_p_t AIRINV::ScheduleParserHelper::class_code_list_p` (`chset_t("A-Z").derived()`, 1, 26)

### Variables

- `int1_p_t AIRINV::ScheduleParserHelper::int1_p`
- `uint2_p_t AIRINV::ScheduleParserHelper::uint2_p`
- `uint4_p_t AIRINV::ScheduleParserHelper::uint4_p`
- `uint1_4_p_t AIRINV::ScheduleParserHelper::uint1_4_p`
- `int1_p_t AIRINV::ScheduleParserHelper::family_code_p`

## 11.80 `airinv/command/ScheduleParserHelper.hpp` File Reference

```
#include <string>
#include <stdair/command/CmdAbstract.hpp>
#include <airinv/AIRINV_Types.hpp>
#include <airinv/basic/BasParserTypes.hpp>
#include <airinv/bom/FlightPeriodStruct.hpp>
```

### Namespaces

- namespace `stdair`
- namespace `AIRINV`
- namespace `AIRINV::ScheduleParserHelper`

### Classes

- struct `AIRINV::ScheduleParserHelper::ParserSemanticAction`
- struct `AIRINV::ScheduleParserHelper::storeAirlineCode`
- struct `AIRINV::ScheduleParserHelper::storeFlightNumber`
- struct `AIRINV::ScheduleParserHelper::storeDateRangeStart`
- struct `AIRINV::ScheduleParserHelper::storeDateRangeEnd`
- struct `AIRINV::ScheduleParserHelper::storeDow`
- struct `AIRINV::ScheduleParserHelper::storeLegBoardingPoint`
- struct `AIRINV::ScheduleParserHelper::storeLegOffPoint`
- struct `AIRINV::ScheduleParserHelper::storeOperatingAirlineCode`
- struct `AIRINV::ScheduleParserHelper::storeOperatingFlightNumber`
- struct `AIRINV::ScheduleParserHelper::storeBoardingTime`
- struct `AIRINV::ScheduleParserHelper::storeOffTime`
- struct `AIRINV::ScheduleParserHelper::storeElapsedTime`
- struct `AIRINV::ScheduleParserHelper::storeLegCabinCode`
- struct `AIRINV::ScheduleParserHelper::storeCapacity`

- struct `AIRINV::ScheduleParserHelper::storeSegmentSpecificity`
- struct `AIRINV::ScheduleParserHelper::storeSegmentBoardingPoint`
- struct `AIRINV::ScheduleParserHelper::storeSegmentOffPoint`
- struct `AIRINV::ScheduleParserHelper::storeSegmentCabinCode`
- struct `AIRINV::ScheduleParserHelper::storeClasses`
- struct `AIRINV::ScheduleParserHelper::storeFamilyCode`
- struct `AIRINV::ScheduleParserHelper::storeFRAT5CurveKey`
- struct `AIRINV::ScheduleParserHelper::storeFFDisutilityCurveKey`
- struct `AIRINV::ScheduleParserHelper::storeFCClasses`
- struct `AIRINV::ScheduleParserHelper::doEndFlight`
- struct `AIRINV::ScheduleParserHelper::FlightPeriodParser`
- struct `AIRINV::ScheduleParserHelper::FlightPeriodParser::definition< ScannerT >`
- class `AIRINV::FlightPeriodFileParser`

## 11.81 `airinv/command/vault/DCPEventGenerator.cpp` File Reference

```
#include <cassert>
#include <stdair/bom/BomManager.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/factory/FacBomManager.hpp>
#include <stdair/service/Logger.hpp>
#include <airinv/bom/DCPEventStruct.hpp>
#include <airinv/command/DCPEventGenerator.hpp>
```

### Namespaces

- namespace `AIRINV`

## 11.82 `airinv/command/vault/DCPEventGenerator.hpp` File Reference

```
#include <stdair/command/CmdAbstract.hpp>
#include <airinv/AIRINV_Types.hpp>
```

### Namespaces

- namespace `stdair`
- namespace `AIRINV`
- namespace `AIRINV::DCPParserHelper`

### Classes

- class `AIRINV::DCPEventGenerator`

## 11.83 airinv/command/vault/DCPParser.cpp File Reference

```
#include <cassert>
#include <string>
#include <stdair/service/Logger.hpp>
#include <airinv/command/DCPParserHelper.hpp>
#include <airinv/command/DCPParser.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.84 airinv/command/vault/DCPParser.hpp File Reference

```
#include <stdair/stdair_basic_types.hpp>
#include <stdair/command/CmdAbstract.hpp>
```

### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

### Classes

- class [AIRINV::DCPParser](#)

## 11.85 airinv/command/vault/DCPParserHelper.cpp File Reference

```
#include <cassert>
#include <string>
#include <vector>
#include <fstream>
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/service/Logger.hpp>
#include <airinv/command/DCPParserHelper.hpp>
#include <airinv/command/DCPRuleGenerator.hpp>
```

### Namespaces

- namespace [AIRINV](#)
- namespace [AIRINV::DCPParserHelper](#)

## Variables

- [stdair::int1\\_p\\_t AIRINV::DCPParserHelper::int1\\_p](#)
- [stdair::uint2\\_p\\_t AIRINV::DCPParserHelper::uint2\\_p](#)
- [stdair::uint4\\_p\\_t AIRINV::DCPParserHelper::uint4\\_p](#)
- [stdair::uint1\\_4\\_p\\_t AIRINV::DCPParserHelper::uint1\\_4\\_p](#)
- [stdair::hour\\_p\\_t AIRINV::DCPParserHelper::hour\\_p](#)
- [stdair::minute\\_p\\_t AIRINV::DCPParserHelper::minute\\_p](#)
- [stdair::second\\_p\\_t AIRINV::DCPParserHelper::second\\_p](#)
- [stdair::year\\_p\\_t AIRINV::DCPParserHelper::year\\_p](#)
- [stdair::month\\_p\\_t AIRINV::DCPParserHelper::month\\_p](#)
- [stdair::day\\_p\\_t AIRINV::DCPParserHelper::day\\_p](#)

## 11.86 airinv/command/vault/DCPParserHelper.hpp File Reference

```
#include <stdair/basic/BasParserTypes.hpp>
#include <stdair/command/CmdAbstract.hpp>
#include <airinv/AIRINV_Types.hpp>
#include <airinv/bom/DCPRuleStruct.hpp>
```

## Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)
- namespace [AIRINV::DCPParserHelper](#)

## Classes

- struct [AIRINV::DCPParserHelper::ParserSemanticAction](#)
- struct [AIRINV::DCPParserHelper::storeDCPID](#)
- struct [AIRINV::DCPParserHelper::storeOrigin](#)
- struct [AIRINV::DCPParserHelper::storeDestination](#)
- struct [AIRINV::DCPParserHelper::storeDateRangeStart](#)
- struct [AIRINV::DCPParserHelper::storeDateRangeEnd](#)
- struct [AIRINV::DCPParserHelper::storeStartRangeTime](#)
- struct [AIRINV::DCPParserHelper::storeEndRangeTime](#)
- struct [AIRINV::DCPParserHelper::storePOS](#)
- struct [AIRINV::DCPParserHelper::storeCabinCode](#)
- struct [AIRINV::DCPParserHelper::storeChannel](#)
- struct [AIRINV::DCPParserHelper::storeAdvancePurchase](#)
- struct [AIRINV::DCPParserHelper::storeSaturdayStay](#)
- struct [AIRINV::DCPParserHelper::storeChangeFees](#)
- struct [AIRINV::DCPParserHelper::storeNonRefundable](#)
- struct [AIRINV::DCPParserHelper::storeMinimumStay](#)
- struct [AIRINV::DCPParserHelper::storeDCP](#)
- struct [AIRINV::DCPParserHelper::storeAirlineCode](#)
- struct [AIRINV::DCPParserHelper::storeClass](#)
- struct [AIRINV::DCPParserHelper::doEndDCP](#)
- struct [AIRINV::DCPParserHelper::DCPRuleParser](#)
- class [AIRINV::DCPRuleFileParser](#)

## 11.87 airinv/config/airinv-paths.hpp.in File Reference

### Defines

- `#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@"`

### 11.87.1 Define Documentation

#### 11.87.1.1 `#define PACKAGE "@PACKAGE@"`

Definition at line 4 of file `airinv-paths.hpp.in`.

#### 11.87.1.2 `#define PACKAGE_NAME "@PACKAGE_NAME@"`

Definition at line 5 of file `airinv-paths.hpp.in`.

#### 11.87.1.3 `#define PACKAGE_VERSION "@PACKAGE_VERSION@"`

Definition at line 6 of file `airinv-paths.hpp.in`.

#### 11.87.1.4 `#define PREFIXDIR "@prefix@"`

Definition at line 7 of file `airinv-paths.hpp.in`.

#### 11.87.1.5 `#define EXEC_PREFIX "@exec_prefix@"`

Definition at line 8 of file `airinv-paths.hpp.in`.

#### 11.87.1.6 `#define BINDIR "@bindir@"`

Definition at line 9 of file `airinv-paths.hpp.in`.

**11.87.1.7 #define LIBDIR "@libdir@"**

Definition at line 10 of file airinv-paths.hpp.in.

**11.87.1.8 #define LIBEXECDIR "@libexecdir@"**

Definition at line 11 of file airinv-paths.hpp.in.

**11.87.1.9 #define SBINDIR "@sbindir@"**

Definition at line 12 of file airinv-paths.hpp.in.

**11.87.1.10 #define SYSCONFDIR "@sysconfdir@"**

Definition at line 13 of file airinv-paths.hpp.in.

**11.87.1.11 #define INCLUDEDIR "@includedir@"**

Definition at line 14 of file airinv-paths.hpp.in.

**11.87.1.12 #define DATAROOTDIR "@datarootdir@"**

Definition at line 15 of file airinv-paths.hpp.in.

**11.87.1.13 #define DATADIR "@datadir@"**

Definition at line 16 of file airinv-paths.hpp.in.

**11.87.1.14 #define DOCDIR "@docdir@"**

Definition at line 17 of file airinv-paths.hpp.in.

**11.87.1.15 #define MANDIR "@mandir@"**

Definition at line 18 of file airinv-paths.hpp.in.

**11.87.1.16 #define INFODIR "@infodir@"**

Definition at line 19 of file airinv-paths.hpp.in.

**11.87.1.17 #define HTMLDIR "@htmldir@"**

Definition at line 20 of file airinv-paths.hpp.in.

**11.87.1.18 #define PDFDIR "@pdfdir@"**

Definition at line 21 of file airinv-paths.hpp.in.

**11.87.1.19 #define STDAIR\_SAMPLE\_DIR "@sampledir@"**

Definition at line 22 of file airinv-paths.hpp.in.



## 11.88 airinv/factory/FacAirinvMasterServiceContext.cpp File Reference

```
#include <cassert>
#include <stdair/service/FacSupervisor.hpp>
#include <airinv/factory/FacAirinvMasterServiceContext.hpp>
#include <airinv/service/AIRINV_Master_ServiceContext.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.89 airinv/factory/FacAirinvMasterServiceContext.hpp File Reference

```
#include <string>
#include <stdair/service/FacServiceAbstract.hpp>
```

### Namespaces

- namespace [AIRINV](#)

### Classes

- class [AIRINV::FacAirinvMasterServiceContext](#)  
*Factory for Bucket.*

## 11.90 airinv/factory/FacAirinvServiceContext.cpp File Reference

```
#include <cassert>
#include <stdair/service/FacSupervisor.hpp>
#include <airinv/factory/FacAirinvServiceContext.hpp>
#include <airinv/service/AIRINV_ServiceContext.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.91 airinv/factory/FacAirinvServiceContext.hpp File Reference

```
#include <string>
#include <stdair/service/FacServiceAbstract.hpp>
```

### Namespaces

- namespace [AIRINV](#)

### Classes

- class [AIRINV::FacAirinvServiceContext](#)

## 11.92 airinv/factory/FacBomAbstract.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <boost/functional/hash/hash.hpp>
#include <airinv/bom/BomAbstract.hpp>
#include <airinv/factory/FacBomAbstract.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.93 airinv/factory/FacBomAbstract.hpp File Reference

```
#include <string>
#include <vector>
```

### Namespaces

- namespace [AIRINV](#)

### Classes

- class [AIRINV::FacBomAbstract](#)

## 11.94 airinv/factory/FacServiceAbstract.cpp File Reference

```
#include <cassert>
#include <airinv/service/ServiceAbstract.hpp>
#include <airinv/factory/FacServiceAbstract.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.95 airinv/factory/FacServiceAbstract.hpp File Reference

```
#include <vector>
```

### Namespaces

- namespace [AIRINV](#)

### Classes

- class [AIRINV::FacServiceAbstract](#)

## 11.96 airinv/factory/FacSupervisor.cpp File Reference

```
#include <cassert>
#include <airinv/factory/FacBomAbstract.hpp>
#include <airinv/factory/FacServiceAbstract.hpp>
#include <airinv/factory/FacSupervisor.hpp>
```

### Namespaces

- namespace [AIRINV](#)

## 11.97 airinv/factory/FacSupervisor.hpp File Reference

```
#include <vector>
```

### Namespaces

- namespace [AIRINV](#)

### Classes

- class [AIRINV::FacSupervisor](#)

## 11.98 airinv/FlightRequestStatus.hpp File Reference

```
#include <string>
#include <stdair/basic/StructAbstract.hpp>
```

### Namespaces

- namespace [AIRINV](#)

### Classes

- struct [AIRINV::FlightRequestStatus](#)

## 11.99 airinv/server/AirInvClient.cpp File Reference

```
#include <string>
#include <iostream>
#include <zmq.hpp>
```

### Functions

- int `main` (int argc, char \*argv[ ])

#### 11.99.1 Function Documentation

##### 11.99.1.1 int main (int *argc*, char \* *argv*[ ])

Definition at line 11 of file AirInvClient.cpp.

## 11.100 airinv/server/AirInvClient\_ASIO.cpp File Reference

```
#include <cassert>
#include <iostream>
#include <string>
#include <boost/asio.hpp>
#include <boost/array.hpp>
```

### Functions

- int `main` (int argc, char \*argv[ ])

#### 11.100.1 Function Documentation

##### 11.100.1.1 int main (int *argc*, char \* *argv*[ ])

Definition at line 14 of file AirInvClient\_ASIO.cpp.

## 11.101 airinv/server/AirInvServer.cpp File Reference

## 11.102 airinv/server/AirInvServer.hpp File Reference

```
#include <string>
#include <vector>
#include <boost/asio.hpp>
#include <boost/noncopyable.hpp>
#include <boost/shared_ptr.hpp>
#include <stdair/stdair_basic_types.hpp>
```

```
#include <airinv/server/Connection.hpp>
#include <airinv/server/RequestHandler.hpp>
```

### Namespaces

- namespace [AIRINV](#)

### Classes

- class [AIRINV::AirInvServer](#)

## 11.103 airinv/server/AirInvServer\_ASIO.cpp File Reference

```
#include <cassert>
#include <boost/thread.hpp>
#include <boost/bind.hpp>
#include <airinv/server/AirInvServer.hpp>
```

### Namespaces

- namespace [AIRINV](#)

### Typedefs

- typedef boost::shared\_ptr< boost::thread > [AIRINV::ThreadShrPtr\\_T](#)
- typedef std::vector< [ThreadShrPtr\\_T](#) > [AIRINV::ThreadShrPtrList\\_T](#)

## 11.104 airinv/server/BomPropertyTree.cpp File Reference

```
#include <boost/property_tree/ptree.hpp>
#include <boost/property_tree/json_parser.hpp>
#include <boost/foreach.hpp>
#include <airinv/server/BomPropertyTree.hpp>
```

### Namespaces

- namespace [stdair](#)

## 11.105 airinv/server/BomPropertyTree.hpp File Reference

```
#include <string>
#include <set>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_date_time_types.hpp>
```

## Namespaces

- namespace [stdair](#)

## Classes

- struct [stdair::BomPropertyTree](#)

## 11.106 airinv/server/Connection.cpp File Reference

```
#include <cassert>
#include <vector>
#include <boost/bind.hpp>
#include <airinv/server/RequestHandler.hpp>
#include <airinv/server/Connection.hpp>
```

## Namespaces

- namespace [AIRINV](#)

## 11.107 airinv/server/Connection.hpp File Reference

```
#include <boost/asio.hpp>
#include <boost/array.hpp>
#include <boost/noncopyable.hpp>
#include <boost/shared_ptr.hpp>
#include <boost/enable_shared_from_this.hpp>
#include <airinv/server/Reply.hpp>
#include <airinv/server/Request.hpp>
```

## Namespaces

- namespace [AIRINV](#)

## Classes

- class [AIRINV::Connection](#)

## Typedefs

- typedef boost::shared\_ptr< Connection > [AIRINV::ConnectionShrPtr\\_T](#)

## 11.108 airinv/server/header.hpp File Reference

```
#include <string>
```

### Namespaces

- namespace [AIRINV](#)

### Classes

- struct [AIRINV::header](#)

## 11.109 airinv/server/posix\_main.cpp File Reference

```
#include <iostream>
#include <string>
#include <boost/asio.hpp>
#include <boost/thread.hpp>
#include <boost/bind.hpp>
#include <boost/lexical_cast.hpp>
#include <airinv/server/AirInvServer.hpp>
#include <pthread.h>
#include <signal.h>
```

### Functions

- int [main](#) (int argc, char \*argv[ ])

#### 11.109.1 Function Documentation

##### 11.109.1.1 int main (int *argc*, char \* *argv*[ ])

Definition at line 25 of file posix\_main.cpp.

References [AIRINV::AirInvServer::run\(\)](#).

## 11.110 airinv/server/Reply.cpp File Reference

```
#include <cassert>
#include <string>
#include <boost/lexical_cast.hpp>
#include <airinv/server/Reply.hpp>
```

### Namespaces

- namespace [AIRINV](#)

### 11.111 airinv/server/Reply.hpp File Reference

```
#include <string>
#include <vector>
#include <boost/asio.hpp>
#include <airinv/FlightRequestStatus.hpp>
```

### Namespaces

- namespace [AIRINV](#)

### Classes

- struct [AIRINV::Reply](#)

### 11.112 airinv/server/Request.cpp File Reference

```
#include <cassert>
#include <airinv/server/Request.hpp>
```

### Namespaces

- namespace [AIRINV](#)

### 11.113 airinv/server/Request.hpp File Reference

```
#include <string>
#include <vector>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_date_time_types.hpp>
```

### Namespaces

- namespace [AIRINV](#)

### Classes

- struct [AIRINV::Request](#)



### 11.114 `airinv/server/RequestHandler.cpp` File Reference

```
#include <cassert>
#include <string>
#include <fstream>
#include <sstream>
#include <boost/lexical_cast.hpp>
#include <airinv/server/Reply.hpp>
#include <airinv/server/Request.hpp>
#include <airinv/server/RequestHandler.hpp>
```

#### Namespaces

- namespace [AIRINV](#)

### 11.115 `airinv/server/RequestHandler.hpp` File Reference

```
#include <string>
#include <boost/noncopyable.hpp>
#include <stdair/stdair_basic_types.hpp>
```

#### Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)

#### Classes

- class [AIRINV::RequestHandler](#)  
*The common handler for all incoming requests.*

### 11.116 `airinv/server/RequestParser.cpp` File Reference

```
#include <cassert>
#include <airinv/server/RequestParser.hpp>
#include <airinv/server/Request.hpp>
```

#### Namespaces

- namespace [AIRINV](#)

## 11.117 airinv/server/RequestParser.hpp File Reference

```
#include <boost/logic/tribool.hpp>
#include <boost/tuple/tuple.hpp>
```

### Namespaces

- namespace [AIRINV](#)

### Classes

- class [AIRINV::RequestParser](#)  
*Parser for incoming requests.*

## 11.118 airinv/server/win\_main.cpp File Reference

```
#include <iostream>
#include <string>
#include <boost/asio.hpp>
#include <boost/bind.hpp>
#include <boost/function.hpp>
#include <boost/lexical_cast.hpp>
#include <airinv/server/AirInvServer.hpp>
```

## 11.119 airinv/service/AIRINV\_Master\_Service.cpp File Reference

```
#include <cassert>
#include <cmath>
#include <boost/make_shared.hpp>
#include <stdair/stdair_json.hpp>
#include <stdair/basic/BasChronometer.hpp>
#include <stdair/basic/EventType.hpp>
#include <stdair/bom/BomKeyManager.hpp>
#include <stdair/bom/SnapshotStruct.hpp>
#include <stdair/bom/RMEventStruct.hpp>
#include <stdair/service/Logger.hpp>
#include <stdair/STDAIR_Service.hpp>
#include <sevmgr/SEVMGR_Service.hpp>
#include <airinv/basic/BasConst_AIRINV_Service.hpp>
#include <airinv/factory/FacAirinvMasterServiceContext.hpp>
```

```
#include <airinv/command/InventoryParser.hpp>
#include <airinv/command/InventoryManager.hpp>
#include <airinv/service/AIRINV_Master_ServiceContext.hpp>
#include <airinv/AIRINV_Service.hpp>
#include <airinv/AIRINV_Master_Service.hpp>
```

#### Namespaces

- namespace [AIRINV](#)

### 11.120 `airinv/service/AIRINV_Master_ServiceContext.cpp` File Reference

```
#include <cassert>
#include <sstream>
#include <airinv/basic/BasConst_AIRINV_Service.hpp>
#include <airinv/service/AIRINV_Master_ServiceContext.hpp>
```

#### Namespaces

- namespace [AIRINV](#)

### 11.121 `airinv/service/AIRINV_Master_ServiceContext.hpp` File Reference

```
#include <string>
#include <boost/shared_ptr.hpp>
#include <stdair/stdair_service_types.hpp>
#include <stdair/bom/Inventory.hpp>
#include <stdair/service/ServiceAbstract.hpp>
#include <sevmgr/SEVMGR_Types.hpp>
#include <airinv/AIRINV_Types.hpp>
```

#### Namespaces

- namespace [AIRINV](#)

#### Classes

- class [AIRINV::AIRINV\\_Master\\_ServiceContext](#)

### 11.122 `airinv/service/AIRINV_Service.cpp` File Reference

```
#include <cassert>
```

```
#include <boost/make_shared.hpp>
#include <stdair/stdair_json.hpp>
#include <stdair/basic/BasChronometer.hpp>
#include <stdair/basic/JSonCommand.hpp>
#include <stdair/basic/PartnershipTechnique.hpp>
#include <stdair/basic/UnconstrainingMethod.hpp>
#include <stdair/basic/OptimisationMethod.hpp>
#include <stdair/bom/BomKeyManager.hpp>
#include <stdair/bom/BomManager.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/bom/Inventory.hpp>
#include <stdair/bom/FlightDate.hpp>
#include <stdair/bom/SegmentCabin.hpp>
#include <stdair/bom/AirlineFeature.hpp>
#include <stdair/bom/RMEventStruct.hpp>
#include <stdair/bom/BomJSONImport.hpp>
#include <stdair/bom/BomJSONExport.hpp>
#include <stdair/factory/FacBomManager.hpp>
#include <stdair/service/Logger.hpp>
#include <stdair/STDAIR_Service.hpp>
#include <rmol/RMOL_Service.hpp>
#include <airrac/AIRRAC_Service.hpp>
#include <sevmgr/SEVMGR_Service.hpp>
#include <airinv/basic/BasConst_AIRINV_Service.hpp>
#include <airinv/factory/FacAirinvServiceContext.hpp>
#include <airinv/command/ScheduleParser.hpp>
#include <airinv/command/FRAT5Parser.hpp>
#include <airinv/command/FFDisutilityParser.hpp>
#include <airinv/command/InventoryParser.hpp>
#include <airinv/command/InventoryManager.hpp>
#include <airinv/command/InventoryBuilder.hpp>
#include <airinv/service/AIRINV_ServiceContext.hpp>
#include <airinv/AIRINV_Service.hpp>
```

## Namespaces

- namespace [AIRINV](#)

### 11.123 airinv/service/AIRINV\_ServiceContext.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <airinv/basic/BasConst_AIRINV_Service.hpp>
#include <airinv/service/AIRINV_ServiceContext.hpp>
```

#### Namespaces

- namespace [AIRINV](#)

### 11.124 airinv/service/AIRINV\_ServiceContext.hpp File Reference

```
#include <string>
#include <boost/shared_ptr.hpp>
#include <stdair/stdair_service_types.hpp>
#include <stdair/service/ServiceAbstract.hpp>
#include <rmol/RMOL_Types.hpp>
#include <airrac/AIRRAC_Types.hpp>
#include <sevmgr/SEVMGR_Types.hpp>
#include <airinv/AIRINV_Types.hpp>
```

#### Namespaces

- namespace [AIRINV](#)

#### Classes

- class [AIRINV::AIRINV\\_ServiceContext](#)  
*Class holding the context of the AirInv services.*

### 11.125 airinv/service/ServiceAbstract.cpp File Reference

```
#include <airinv/service/ServiceAbstract.hpp>
```

#### Namespaces

- namespace [AIRINV](#)

### 11.126 airinv/service/ServiceAbstract.hpp File Reference

```
#include <iosfwd>
```

## Namespaces

- namespace [AIRINV](#)

## Classes

- class [AIRINV::ServiceAbstract](#)

## Functions

- `template<class charT, class traits> std::basic_ostream< charT, traits > & operator<< (std::basic_ostream< charT, traits > &ioOut, const AIRINV::ServiceAbstract &iService)`
- `template<class charT, class traits> std::basic_istream< charT, traits > & operator>> (std::basic_istream< charT, traits > &ioIn, AIRINV::ServiceAbstract &ioService)`

### 11.126.1 Function Documentation

**11.126.1.1** `template<class charT, class traits> std::basic_ostream<charT, traits>& operator<< (std::basic_ostream< charT, traits > & ioOut, const AIRINV::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.

**11.126.1.2** `template<class charT, class traits> std::basic_istream<charT, traits>& operator>> (std::basic_istream< charT, traits > & ioIn, AIRINV::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 [AIRINV::ServiceAbstract::fromStream\(\)](#).

- 11.127 [airinv/ui/cmdline/airinv.cpp](#) File Reference
- 11.128 [doc/local/authors.doc](#) File Reference
- 11.129 [doc/local/codingrules.doc](#) File Reference
- 11.130 [doc/local/copyright.doc](#) File Reference
- 11.131 [doc/local/documentation.doc](#) File Reference
- 11.132 [doc/local/features.doc](#) File Reference
- 11.133 [doc/local/help\\_wanted.doc](#) File Reference
- 11.134 [doc/local/howto\\_release.doc](#) File Reference
- 11.135 [doc/local/index.doc](#) File Reference
- 11.136 [doc/local/installation.doc](#) File Reference
- 11.137 [doc/local/linking.doc](#) File Reference
- 11.138 [doc/local/test.doc](#) File Reference
- 11.139 [doc/local/users\\_guide.doc](#) File Reference
- 11.140 [doc/local/verification.doc](#) File Reference
- 11.141 [doc/tutorial/tutorial.doc](#) File Reference
- 11.142 [test/airinv/InventoryTestSuite.cpp](#) File Reference
- 11.143 [test/airinv/InventoryTestSuite.hpp](#) File Reference

```
#include <iosfwd>
```

```
#include <cppunit/extensions/HelperMacros.h>
```

#### Classes

- class [InventoryTestSuite](#)

#### Functions

- [CPPUNIT\\_TEST\\_SUITE\\_REGISTRATION](#) ([InventoryTestSuite](#))

#### 11.143.1 Function Documentation

##### 11.143.1.1 [CPPUNIT\\_TEST\\_SUITE\\_REGISTRATION](#) ([InventoryTestSuite](#))

## 12 AirInv Page Documentation

### 12.1 People

#### 12.1.1 Project Admins

- Denis Arnaud <denis\_arnaud@users.sourceforge.net> ([N](#))
- Anh Quan Nguyen <quannaus@users.sourceforge.net> ([N](#))

#### 12.1.2 Developers

- Anh Quan Nguyen <quannaus@users.sourceforge.net> ([N](#))
- Denis Arnaud <denis\_arnaud@users.sourceforge.net> ([N](#))
- Son Nguyen Kim <snguyenkim@users.sourceforge.net> ([N](#))
- Nicolas Bondoux <nbondoux@users.sourceforge.net> ([N](#))

#### 12.1.3 Retired Developers

- Patrick Grandjean <pgrandjean@users.sourceforge.net> ([N](#))
- Ngoc-Thach Hoang <hoangngocthach@users.sourceforge.net> ([N](#))

#### 12.1.4 Contributors

- Emmanuel Bastien <ebastien@users.sourceforge.net> ([N](#))
- Christophe Lacombe <ddtof@users.sourceforge.net> ([N](#))

#### 12.1.5 Distribution Maintainers

- **Fedora/RedHat**: Denis Arnaud <denis\_arnaud@users.sourceforge.net> ([N](#))
- **Debian**: Emmanuel Bastien <ebastien@users.sourceforge.net> ([N](#))

#### Note:

(N) - **Amadeus** employees.

### 12.2 Coding Rules

In the following sections we describe the naming conventions which are used for files, classes, structures, local variables, and global variables.

#### 12.2.1 Default Naming Rules for Variables

Variables names follow Java naming conventions. Examples:

- `lNumberOfPassengers`
- `lSeatAvailability`



### 12.2.2 Default Naming Rules for Functions

Function names follow Java naming conventions. Example:

- `int myFunctionName (const int& a, int b)`

### 12.2.3 Default Naming Rules for Classes and Structures

Each new word in a class or structure name should always start with a capital letter and the words should be separated with an under-score. Abbreviations are written with capital letters. Examples:

- `MyClassName`
- `MyStructName`

### 12.2.4 Default Naming Rules for Files

Files are named after the C++ class names.

Source files are named using `.cpp` suffix, whereas header files end with `.hpp` extension. Examples:

- `FlightDate.hpp`
- `SegmentDate.cpp`

### 12.2.5 Default Functionality of Classes

All classes that are configured by input parameters should include:

- default empty constructor
- one or more additional constructor(s) that takes input parameters and initializes the class instance
- setup function, preferably named `'setup'` or `'set_parameters'`

Explicit destructor functions are not required, unless they are needed. It shall not be possible to use any of the other member functions unless the class has been properly initiated with the input parameters.

## 12.3 Copyright and License

### 12.3.1 GNU LESSER GENERAL PUBLIC LICENSE

#### 12.3.1.1 Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2, hence  
the version number 2.1.]

### 12.3.2 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages—typically libraries—of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the

Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

### 12.3.3 TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small

macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library,

and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted

by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

**12.3.3.1 NO WARRANTY** 15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

### 12.3.3.2 END OF TERMS AND CONDITIONS

### 12.3.4 How to Apply These Terms to Your New Programs

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
```

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!

Source

## 12.4 Documentation Rules

### 12.4.1 General Rules

All classes in `AirInv` 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 AirInv is shown here:

```

/ *!
 * \brief Brief description of MyClass here
 *
 * Detailed description of MyClass here. With example code if needed.
 */
class MyClass {
public:
    ///! Default constructor
    MyClass(void) { setup_done = false; }

    / *!
     * \brief Constructor that initializes the class with parameters
     *
     * Detailed description of the constructor here if needed
     *
     * \param[in] param1 Description of \a param1 here
     * \param[in] param2 Description of \a param2 here
     */
    MyClass(TYPE1 param1, TYPE2 param2) { setup(param1, param2); }

    / *!
     * \brief Setup function for MyClass
     *
     * Detailed description of the setup function here if needed
     *
     * \param[in] param1 Description of \a param1 here
     * \param[in] param2 Description of \a param2 here
     */
    void setup(TYPE1 param1, TYPE2 param2);

    / *!
     * \brief Brief description of memberFunction1
     *
     * Detailed description of memberFunction1 here if needed
     *
     * \param[in] param1 Description of \a param1 here
     * \param[in] param2 Description of \a param2 here
     * \param[in,out] param3 Description of \a param3 here
     * \return Description of the return value here
     */
    TYPE4 memberFunction1(TYPE1 param1, TYPE2 param2, TYPE3 &param3);

private:
    bool _setupDone;          /*!< Variable that checks if the class is properly

```



```

                                initialized with parameters */
TYPE1 _privateVariable1; ///< Short description of _privateVariable1 here
TYPE2 _privateVariable2; ///< Short description of _privateVariable2 here
};

```

### 12.4.2 File Header

All files should start with the following header, which include Doxygen's `\file`, `\brief` and `\author` tags, `$Date$` and `$Revisions$` CVS tags, and a common copyright note:

```

/ *!
 * \file
 * \brief Brief description of the file here
 * \author Names of the authors who contributed to this code
 * \date Date
 *
 * Detailed description of the file here if needed.
 *
 * -----
 *
 * AirInv - C++ Airline Inventory Management Library
 *
 * Copyright (C) 2009-2010 (\see authors file for a list of contributors)
 *
 * \see copyright file for license information
 *
 * -----
 */

```

### 12.4.3 Grouping Various Parts

All functions must be added to a Doxygen group in order to appear in the documentation. The following code example defines the group `'my_group'`:

```

/ *!
 * \defgroup my_group Brief description of the group here
 *
 * Detailed description of the group here
 */

```

The following example shows how to document the function `myFunction` and how to add it to the group `my_group`:

```

/ *!
 * \brief Brief description of myFunction here
 * \ingroup my_group
 *
 * Detailed description of myFunction here
 *
 * \param[in] param1 Description of \a param1 here
 * \param[in] param2 Description of \a param2 here
 * \return Description of the return value here
 */
TYPE3 myFunction(TYPE1 param1, TYPE2 &param2);

```

## 12.5 Main features

A short list of the main features of AirInv is given below sorted in different categories. Many more features and functions exist and for these we refer to the reference documentation.

### 12.5.1 Network generation

- Network/graph generation

### 12.5.2 Inventory generation

- Inventory generation

### 12.5.3 Finding travel solutions

- Matching of travel solutions with user requests

### 12.5.4 Distributed inventories

- Inventory independent partitions
- MPI-based distribution

### 12.5.5 Other features

- CSV input file parsing
- Memory handling

## 12.6 Make a Difference

**Do not ask what AirSched can do for you. Ask what you can do for AirSched.**

You can help us to develop the AirSched library. There are always a lot of things you can do:

- Start using AirSched
- Tell your friends about AirSched 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 AirSched discussion forums on SourceForge. If you know the answer to a question, help others to overcome their AirSched 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 AirSched to new platforms. If you manage to compile AirSched on a new platform, then tell us how you did it.
- Send us your code. If you have a good AirSched compatible code, which you can release under the LGPLv2.1, and you think it should be included in AirSched, then send it to us.
- Become an AirSched developer. Send us an e-mail and tell what you can do for AirSched.

## 12.7 Make a new release

### 12.7.1 Introduction

This document describes briefly the recommended procedure of releasing a new version of AirInv using a Linux development machine and the SourceForge project site.

The following steps are required to make a release of the distribution package.

### 12.7.2 Initialisation

Clone locally the full [Git project](#):

```
cd ~
mkdir -p dev/sim
cd ~/dev/sim
git clone git://airinv.git.sourceforge.net/gitroot/airinv/airinv airinvgit
cd airinvgit
git checkout trunk
```

### 12.7.3 Branch creation

Create the branch, on your local clone, corresponding to the new release (say, 0.5.0):

```
cd ~/dev/sim/airinvgit
git checkout trunk
git checkout -b 0.5.0
```

Update the version in the various build system files, replacing 99.99.99 by the correct version number:

```
vi CMakeLists.txt
vi autogen.sh
```

Update the version and add a change-log in the ChangeLog and in the RPM specification files:

```
vi ChangeLog
vi airinv.spec
```

### 12.7.4 Commit and publish the release branch

Commit the new release:

```
cd ~/dev/sim/airinvgit
git add -A
git commit -m "[Release 0.5.0] Release of version 0.5.0."
git push
```

### 12.7.5 Update the change-log in the trunk as well

Update the change-log in the ChangeLog and RPM specification files:

```
cd ~/dev/sim/airinvgit
git checkout trunk
vi ChangeLog
vi airinv.spec
```

Commit the change-logs and publish the trunk (main development branch):

```
git commit -m "[Doc] Integrated the change-log of the release 0.5.0."
git push
```

### 12.7.6 Create distribution packages

Create the distribution packages using the following command:

```
cd ~/dev/sim/airinvgit
git checkout 0.5.0
rm -rf build && mkdir -p build
cd build
cmake -DCMAKE_INSTALL_PREFIX=/home/user/dev/deliveries/airinv-0.5.0 \
      -DWITH_STDAIR_PREFIX=/home/user/dev/deliveries/stdair-stable \
      -DCMAKE_BUILD_TYPE:STRING=Debug -DINSTALL_DOC:BOOL=ON ..
make check && make dist
```

This will configure, compile and check the package. The output packages will be named, for instance, `airinv-0.5.0.tar.gz` and `airinv-0.5.0.tar.bz2`.

### 12.7.7 Generation the RPM packages

Optionally, generate the RPM package (for instance, for [Fedora/RedHat](#)):

```
cd ~/dev/sim/airinvgit
git checkout 0.5.0
rm -rf build && mkdir -p build
cd build
cmake -DCMAKE_INSTALL_PREFIX=/home/user/dev/deliveries/airinv-0.5.0 \
      -DWITH_STDAIR_PREFIX=/home/user/dev/deliveries/stdair-stable \
      -DCMAKE_BUILD_TYPE:STRING=Debug -DINSTALL_DOC:BOOL=ON ..
make dist
```

To perform this step, `rpm-build`, `rpmlint` and `rpmdevtools` have to be available on the system.

```
cp airinv.spec ~/dev/packages/SPECS \
  && cp airinv-0.5.0.tar.bz2 ~/dev/packages/SOURCES
cd ~/dev/packages/SPECS
rpmbuild -ba airinv.spec
rpmlint -i ../SPECS/airinv.spec ../SRPMS/airinv-0.5.0-1.fc15.src.rpm \
  ../RPMS/noarch/airinv-* ../RPMS/i686/airinv-*
```

### 12.7.8 Update distributed change log

Update the `NEWS` and `ChangeLog` files with appropriate information, including what has changed since the previous release. Then commit and push the changes into the [AirInv's Git repository](#).

### 12.7.9 Create the binary package, including the documentation

Create the binary package, which includes HTML and PDF documentation, using the following command:

```
make package
```

The output binary package will be named, for instance, `airinv-0.5.0-Linux.tar.bz2`. That package contains both the HTML and PDF documentation. The binary package contains also the executables and shared libraries, as well as C++ header files, but all of those do not interest us for now.

### 12.7.10 Upload the files to SourceForge

Upload the distribution and documentation packages to the SourceForge server. Check [SourceForge help page on uploading software](#).

### 12.7.11 Upload the documentation to SourceForge

In order to update the Web site files, either:

- [synchronise them with rsync and SSH](#):

```
cd ~/dev/sim/airinvgit
git checkout 0.5.0
rsync -aiv doc/html/ doc/latex/refman.pdf joe,airinv@web.sourceforge.net:htdocs/
```

where `-aiv` options mean:

- `-a`: archive/mirror mode; equals `-rlptgoD` (no `-H`, `-A`, `-X`)
  - `-v`: increase verbosity
  - `-i`: output a change-summary for all updates
  - Note the trailing slashes (/) at the end of both the source and target directories. It means that the content of the source directory (`doc/html`), rather than the directory itself, has to be copied into the content of the target directory.
- or use the [SourceForge Shell service](#).

### 12.7.12 Make a new post

- submit a new entry in the [SourceForge project-related news feed](#)
- make a new post on the [SourceForge hosted WordPress blog](#)
- and update, if necessary, [Trac tickets](#).

### 12.7.13 Send an email on the announcement mailing-list

Finally, you should send an announcement to [airinv-announce@lists.sourceforge.net](mailto:airinv-announce@lists.sourceforge.net) (see <https://lists.sourceforge.net/lists/listinfo/airinv-announce> for the archives)

## 12.8 Installation

### 12.8.1 Table of Contents

- [Fedora/RedHat Linux distributions](#)
- [Airinv Requirements](#)
- [Basic Installation](#)
- [Compilers and Options](#)
- [Compiling For Multiple Architectures](#)

- [Installation Names](#)
- [Optional Features](#)
- [Particular systems](#)
- [Specifying the System Type](#)
- [Sharing Defaults](#)
- [Defining Variables](#)
- [‘cmake’ Invocation](#)

### 12.8.2 Fedora/RedHat Linux distributions

Note that on [Fedora/RedHat](#) Linux distributions, RPM packages are available and can be installed with your usual package manager. For instance:

```
yum -y install airinv-devel airinv-doc
```

RPM packages can also be available on the [SourceForge download site](#).

### 12.8.3 Airinv Requirements

Airinv 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``)
- [SOXI](#) - 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 Airinv.

### 12.8.4 Basic Installation

Briefly, the shell commands `./cmake .. && make install` should configure, build, and install this package. The following more-detailed instructions are generic; see the `'README'` file for instructions specific to this package. Some packages provide this `'INSTALL'` file but do not implement all of the features documented below. The lack of an optional feature in a given package is not necessarily a bug. More recommendations for GNU packages can be found in the info page corresponding to "Makefile Conventions: (standards)Makefile Conventions".

The `'cmake'` shell script attempts to guess correct values for various system-dependent variables used during compilation. It uses those values to create a `'Makefile'` in each directory of the package. It may also create one or more `'.h'` files containing system-dependent definitions. Finally, it creates a `'CMakeCache.txt'` cache file that you can refer to in the future to recreate the current configuration, and 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.

### 12.8.5 Compilers and Options

Some systems require unusual options for compilation or linking that the 'cmake' script does not know about. Run './cmake -help' for details on some of the pertinent environment variables.

You can give 'cmake' initial values for configuration parameters by setting variables in the command line or in the environment. Here is an example:

```
./cmake CC=c99 CFLAGS=-g LIBS=-lposix
```

**See also:**

[Defining Variables](#) for more details.

### 12.8.6 Compiling For Multiple Architectures

You can compile the package for more than one kind of computer at the same time, by placing the object files for each architecture in their own directory. To do this, you can use GNU 'make'. 'cd' to the directory where you want the object files and executables to go and run the 'configure' script. 'configure' automatically checks for the source code in the directory that 'configure' is in and in '..'. This is known as a "VPATH" build.

With a non-GNU 'make', it is safer to compile the package for one architecture at a time in the source code directory. After you have installed the package for one architecture, use 'make distclean' before reconfiguring for another architecture.

On MacOS X 10.5 and later systems, you can create libraries and executables that work on multiple system types-known as "fat" or "universal" binaries-by specifying multiple '-arch' options to the compiler but only a single '-arch' option to the preprocessor. Like this:

```
./configure CC="gcc -arch i386 -arch x86_64 -arch ppc -arch ppc64" \  
CXX="g++ -arch i386 -arch x86_64 -arch ppc -arch ppc64" \  
CPP="gcc -E" CXXCPP="g++ -E"
```

This is not guaranteed to produce working output in all cases, you may have to build one architecture at a time and combine the results using the 'lipo' tool if you have problems.

### 12.8.7 Installation Names

By default, 'make install' installs the package's commands under '/usr/local/bin', include files under '/usr/local/include', etc. You can specify an installation prefix other than '/usr/local' by giving 'configure' the option '-prefix=PREFIX', where PREFIX must be an absolute file name.

You can specify separate installation prefixes for architecture-specific files and architecture-independent files. If



you pass the option `'-exec-prefix=PREFIX'` to `'configure'`, the package uses PREFIX as the prefix for installing programs and libraries. Documentation and other data files still use the regular prefix.

In addition, if you use an unusual directory layout you can give options like `'-bindir=DIR'` to specify different values for particular kinds of files. Run `'configure -help'` for a list of the directories you can set and what kinds of files go in them. In general, the default for these options is expressed in terms of `'${prefix}'`, so that specifying just `'-prefix'` will affect all of the other directory specifications that were not explicitly provided.

The most portable way to affect installation locations is to pass the correct locations to `'configure'`; however, many packages provide one or both of the following shortcuts of passing variable assignments to the `'make install'` command line to change installation locations without having to reconfigure or recompile.

The first method involves providing an override variable for each affected directory. For example, `'make install prefix=/alternate/directory'` will choose an alternate location for all directory configuration variables that were expressed in terms of `'${prefix}'`. Any directories that were specified during `'configure'`, but not in terms of `'${prefix}'`, must each be overridden at install time for the entire installation to be relocated. The approach of makefile variable overrides for each directory variable is required by the GNU Coding Standards, and ideally causes no recompilation. However, some platforms have known limitations with the semantics of shared libraries that end up requiring recompilation when using this method, particularly noticeable in packages that use GNU Libtool.

The second method involves providing the `'DESTDIR'` variable. For example, `'make install DESTDIR=/alternate/directory'` will prepend `'/alternate/directory'` before all installation names. The approach of `'DESTDIR'` overrides is not required by the GNU Coding Standards, and does not work on platforms that have drive letters. On the other hand, it does better at avoiding recompilation issues, and works well even when some directory options were not specified in terms of `'${prefix}'` at `'configure'` time.

### 12.8.8 Optional Features

If the package supports it, you can cause programs to be installed with an extra prefix or suffix on their names by giving `'cmake'` the option `'-program-prefix=PREFIX'` or `'-program-suffix=SUFFIX'`.

Some packages pay attention to `'-enable-FEATURE'` options to `'configure'`, where FEATURE indicates an optional part of the package. They may also pay attention to `'-with-PACKAGE'` options, where PACKAGE is something like `'gnu-as'` or `'x'` (for the X Window System). The `'README'` should mention any `'-enable-'` and `'-with-'` options that the package recognizes.

For packages that use the X Window System, `'configure'` can usually find the X include and library files automatically, but if it doesn't, you can use the `'configure'` options `'-x-includes=DIR'` and

`'-x-libraries=DIR'` to specify their locations.

Some packages offer the ability to configure how verbose the execution of `'make'` will be. For these packages, running `'./configure -enable-silent-rules'` sets the default to minimal output, which can be overridden with `'make V=1'`; while running `'./configure -disable-silent-rules'` sets the default to verbose, which can be overridden with `'make V=0'`.

### 12.8.9 Particular systems

On HP-UX, the default C compiler is not ANSI C compatible. If GNU CC is not installed, it is recommended to use the following options in order to use an ANSI C compiler:

```
./configure CC="cc -Ae -D_XOPEN_SOURCE=500"
```

and if that doesn't work, install pre-built binaries of GCC for HP-UX.

On OSF/1 a.k.a. Tru64, some versions of the default C compiler cannot parse its `'<wchar.h>'` header file. The option `'-nodtk'` can be used as a workaround. If GNU CC is not installed, it is therefore recommended to try

```
./configure CC="cc"
```

and if that doesn't work, try

```
./configure CC="cc -nodtk"
```

On Solaris, don't put `'/usr/ucb'` early in your `'PATH'`. This directory contains several dysfunctional programs; working variants of these programs are available in `'/usr/bin'`. So, if you need `'/usr/ucb'` in your `'PATH'`, put it `_after_` `'/usr/bin'`.

On Haiku, software installed for all users goes in `'/boot/common'`, not `'/usr/local'`. It is recommended to use the following options:

```
./cmake -DCMAKE_INSTALL_PREFIX=/boot/common
```

### 12.8.10 Specifying the System Type

There may be some features `'configure'` cannot figure out automatically, but needs to determine by the type of machine the package will run on. Usually, assuming the package is built to be run on the `_same_` architectures, `'configure'` can figure that out, but if it prints a message saying it cannot guess the machine type, give it the `'-build=TYPE'` option. TYPE can either be a short name for the system type, such as `'sun4'`, or a canonical name which has the form CPU-COMPANY-SYSTEM

where SYSTEM can have one of these forms:

- OS

- KERNEL-OS

See the file `'config.sub'` for the possible values of each field. If `'config.sub'` isn't included in this package, then this package doesn't need to know the machine type.

If you are `_building_` compiler tools for cross-compiling, you should use the option `'-target=TYPE'` to select the type of system they will produce code for.

If you want to `_use_` a cross compiler, that generates code for a platform different from the build platform, you should specify the "host" platform (i.e., that on which the generated programs will eventually be run) with `'-host=TYPE'`.

### 12.8.11 Sharing Defaults

If you want to set default values for `'configure'` scripts to share, you can create a site shell script called `'config.site'` that gives default values for variables like `'CC'`, `'cache_file'`, and `'prefix'`. `'configure'` looks for `'PREFIX/share/config.site'` if it exists, then `'PREFIX/etc/config.site'` if it exists. Or, you can set the `'CONFIG_SITE'` environment variable to the location of the site script. A warning: not all `'configure'` scripts look for a site script.

### 12.8.12 Defining Variables

Variables not defined in a site shell script can be set in the environment passed to `'configure'`. However, some packages may run `configure` again during the build, and the customized values of these variables may be lost. In order to avoid this problem, you should set them in the `'configure'` command line, using `'VAR=value'`. For example:

```
./configure CC=/usr/local2/bin/gcc
```

causes the specified `'gcc'` to be used as the C compiler (unless it is overridden in the site shell script).

Unfortunately, this technique does not work for `'CONFIG_SHELL'` due to an Autoconf bug. Until the bug is fixed you can use this workaround:

```
CONFIG_SHELL=/bin/bash /bin/bash ./configure CONFIG_SHELL=/bin/bash
```

### 12.8.13 'cmake' Invocation

`'cmake'` recognizes the following options to control how it operates.

- `'-help'`, `'-h'` print a summary of all of the options to `'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:

```
export LIBSUFFIX_4_CMAKE="-DLIB_SUFFIX=64"
export INSTALL_BASEDIR=/home/user/dev/deliveries
cmake -DCMAKE_INSTALL_PREFIX=${INSTALL_BASEDIR}/airinv-0.5.0 \
  -DWITH_STDAIR_PREFIX=${INSTALL_BASEDIR}/stdair-stable \
  -DWITH_AIRAC_PREFIX=${INSTALL_BASEDIR}/airrac-stable \
  -DWITH_RMOL_PREFIX=${INSTALL_BASEDIR}/rmol-stable \
  -DCMAKE_BUILD_TYPE:String=Debug -DINSTALL_DOC:BOOL=ON ${LIBSUFFIX_4_CMAKE} ..
-- 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: 0ee8dcc3e3dd1d1d442c4054fbfa4cacc1182e6a 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 Readline without specifying any version
-- Found Readline: /usr/include
-- Found Readline version: 6.2
-- 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-0.37
-- 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 'airraclib' to CXX
-- Had to set the linker language for 'rmollib' to CXX
-- Had to set the linker language for 'airinvlib' to CXX
-- Test 'InventoryTestSuite' to be built with 'InventoryTestSuite.cpp'
--
-- =====
-- -----
-- ---      Project Information      ---
-- -----
-- PROJECT_NAME ..... : airinv
-- PACKAGE_PRETTY_NAME ..... : AirInv
-- PACKAGE ..... : airinv
-- PACKAGE_NAME ..... : AIRINV
-- PACKAGE_BRIEF ..... : C++ Simulated Airline Inventory Management System library
-- PACKAGE_VERSION ..... : 0.5.0
-- GENERIC_LIB_VERSION ..... : 0.5.0
-- GENERIC_LIB_SOVERSION ..... : 0.5
--
-- -----
-- ---      Build Configuration      ---
-- -----
-- Modules to build ..... : airrac;rmol;airinv
-- Libraries to build/install ..... : airraclib;rmollib;airinvlib
-- Binaries to build/install ..... : airrac;rmol;airinv_parseInventory;airinv
-- Modules to test ..... : airinv
-- Binaries to test ..... : InventoryTestSuitetst
--
-- * Module ..... : airrac
--   + Layers to build ..... : .;basic;bom;factory;command;service
--   + Dependencies on other layers :
--   + Libraries to build/install . : airraclib
--   + Executables to build/install : airrac
--   + Tests to perform ..... :
-- * Module ..... : rmol
--   + Layers to build ..... : .;basic;bom;factory;command;service
--   + Dependencies on other layers : airraclib
--   + Libraries to build/install . : rmollib
--   + Executables to build/install : rmol
--   + Tests to perform ..... :
-- * Module ..... : airinv
--   + Layers to build ..... : .;basic;bom;factory;command;service
--   + Dependencies on other layers : airraclib;rmollib
--   + Libraries to build/install . : airinvlib
--   + Executables to build/install : airinv_parseInventory;airinv
--   + Tests to perform ..... : InventoryTestSuitetst
--

```

```

-- BUILD_SHARED_LIBS ..... : ON
-- CMAKE_BUILD_TYPE ..... : Debug
-- * CMAKE_C_FLAGS ..... :
-- * CMAKE_CXX_FLAGS ..... : -Wall -Werror
-- * BUILD_FLAGS ..... :
-- * COMPILE_FLAGS ..... :
-- CMAKE_MODULE_PATH ..... : /home/dan/dev/sim/airinv/airinvgithub/config/
-- CMAKE_INSTALL_PREFIX ..... : /home/dan/dev/deliveries/airinv-0.5.0
--
-- * 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/dan/dev/deliveries/airinv-0.5.0/lib64
-- INSTALL_BIN_DIR ..... : /home/dan/dev/deliveries/airinv-0.5.0/bin
-- INSTALL_INCLUDE_DIR ..... : /home/dan/dev/deliveries/airinv-0.5.0/include
-- INSTALL_DATA_DIR ..... : /home/dan/dev/deliveries/airinv-0.5.0/share
-- INSTALL_SAMPLE_DIR ..... : /home/dan/dev/deliveries/airinv-0.5.0/share/airinv/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 ..... : 0.5.0
-- CPACK_PACKAGE_DESCRIPTION_FILE . : /home/dan/dev/sim/airinv/airinvgithub/README
-- CPACK_RESOURCE_FILE_LICENSE .... : /home/dan/dev/sim/airinv/airinvgithub/COPYING
-- CPACK_GENERATOR ..... : TBZ2
-- CPACK_DEBIAN_PACKAGE_DEPENDS ... :
-- CPACK_SOURCE_GENERATOR ..... : TBZ2;TGZ
-- CPACK_SOURCE_PACKAGE_FILE_NAME . : airinv-0.5.0
--
-- -----
-- --- 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_
--
-- * Readline:
--   - READLINE_VERSION ..... : 6.2
--   - READLINE_INCLUDE_DIR ..... : /usr/include
--   - READLINE_LIBRARY ..... : /usr/lib64/libreadline.so
--
-- * 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/dan/dev/deliveries/stdair-0.38.0/bin
--   - STDAIR_EXECUTABLES ..... : stdair
--   - STDAIR_LIBRARY_DIRS ..... : /home/dan/dev/deliveries/stdair-0.38.0/lib64
--   - STDAIR_LIBRARIES ..... : stdairlib;stdairuicllib
--   - STDAIR_INCLUDE_DIRS ..... : /home/dan/dev/deliveries/stdair-0.38.0/include
--   - STDAIR_SAMPLE_DIR ..... : /home/dan/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/dan/dev/sim/airinv/airinvgithub/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_airinv
[ 0%] Built target hdr_cfg_airrac
[ 13%] Built target airraclib
[ 13%] Built target hdr_cfg_rmol
[ 38%] Built target rmolllib
[ 98%] Built target airinvlib
[100%] Built target InventoryTestSuitetst
Scanning dependencies of target check_airinvtst
Test project /home/dan/dev/sim/airinv/airinvgithub/build/test/airinv
  Start 1: InventoryTestSuitetst
1/1 Test #1: InventoryTestSuitetst ..... Passed    0.08 sec

100% tests passed, 0 tests failed out of 1

Total Test time (real) = 0.35 sec
[100%] Built target check_airinvtst
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/airinvgit
rm -rf build && mkdir build
cd build
```

to remove everything.

## 12.9 Linking with Airinv

### 12.9.1 Table of Contents

- [Introduction](#)
- [Dependencies](#)
- [Using the pkg-config command](#)
- [Using the airinv-config script](#)
- [M4 macro for the GNU Autotools](#)
- [Using Airinv with dynamic linking](#)

### 12.9.2 Introduction

There are two convenient methods of linking your programs with the Airinv library. The first one employs the 'pkg-config' command (see <http://pkgconfig.freedesktop.org/>), whereas the second one uses 'airinv-config' script. These methods are shortly described below.

### 12.9.3 Dependencies

The Airinv library depends on several other C++ components.

**12.9.3.1 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: Airinv Dependencies

#### 12.9.4 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 Airinv based program 'my\_prog.cpp', you should use the following command:

```
g++ `pkg-config --cflags airinv` -o my_prog my_prog.cpp `pkg-config --libs airinv`
```

For more information see the 'pkg-config' man pages.

#### 12.9.5 Using the airinv-config script

Airinv provides a shell script called 'airinv-config', which is installed by default in '\$prefix/bin' ('/usr/local/bin') directory. It can be used to simplify compilation and linking of Airinv 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++ `airinv-config --cflags` -o my_prog_opt my_prog.cpp `airinv-config --libs`
```

A list of `'airinv-config'` options can be obtained by typing:

```
airinv-config --help
```

If the `'airinv-config'` command is not found by your shell, you should add its location `'$prefix/bin'` to the `PATH` environment variable, e.g.:

```
export PATH=/usr/local/bin:$PATH
```

### 12.9.6 M4 macro for the GNU Autotools

A M4 macro file is delivered with Airinv, namely `'airinv.m4'`, which can be found in, e.g., `'/usr/share/aclocal'`. When used by a `'configure'` script, thanks to the `'AM_PATH_Airinv'` macro (specified in the M4 macro file), the following Makefile variables are then defined:

- `'Airinv_VERSION'` (e.g., defined to 0.23.0)
- `'Airinv_CFLAGS'` (e.g., defined to `'-I${prefix}/include'`)
- `'Airinv_LIBS'` (e.g., defined to `'-L${prefix}/lib -lairinv'`)

### 12.9.7 Using Airinv with dynamic linking

When using static linking some of the library routines in Airinv 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 Airinv library file during your program execution. If you install the Airinv 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=<Airinv installation prefix>/lib:$LD_LIBRARY_PATH
```

## 12.10 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.

### 12.10.1 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'`.

### 12.10.2 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.

### 12.10.3 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.

## 12.11 Users Guide

### 12.11.1 Table of Contents

- [Introduction](#)
- [Get Started](#)
  - [Get the AirInv library](#)
  - [Build the AirInv project](#)
  - [Build and Run the Tests](#)
  - [Install the AirInv Project \(Binaries, Documentation\)](#)
- [Input file of AirInv 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](#)

### 12.11.2 Introduction

The `AirInv` library contains classes for airline business management. This document does not cover all the aspects of the `AirInv` library. It does however explain the most important things you need to know in order to start using `AirInv`.

### 12.11.3 Get Started

#### 12.11.3.1 Get the AirInv library Clone locally the full [Git project](#):

```
cd ~
mkdir -p dev/sim
cd ~/dev/sim
git clone git://airinv.git.sourceforge.net/gitroot/airinv/airinv airinvgit
cd airinvgit
git checkout trunk
```

#### 12.11.3.2 Build the AirInv project Link with StdAir, create the distribution package (say, 0.5.0) and compile using the following commands:

```
cd ~/dev/sim/airinvgit
rm -rf build && mkdir -p build
cd build
cmake -DCMAKE_INSTALL_PREFIX=~/dev/deliveries/airinv-0.5.0 \
      -DWITH_STDAIR_PREFIX=~/dev/deliveries/stdair-stable \
      -DCMAKE_BUILD_TYPE:STRING=Debug -DINSTALL_DOC:BOOL=ON ..
make
```

#### 12.11.3.3 Build and Run the Tests After building the AirInv project, the following commands run the tests:

```
cd ~/dev/sim/airinvgit
cd build
make check
```

As a result, you should obtain a similar report:

```
[ 0%] Built target hdr_cfg_airinv
[ 96%] Built target airinvlib
[100%] Built target AirlineScheduleTestSuitetst
Scanning dependencies of target check_airinvtst
Test project /home/dan/dev/sim/airinv/airinvgithub/build/test/airinv
  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_airinvtst
Scanning dependencies of target check
[100%] Built target check
```

#### 12.11.3.4 Install the AirInv Project (Binaries, Documentation) After the step [Build the AirInv project](#), to install the library and its header files, type:

```
cd ~/dev/sim/airinvgit
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/airinv-0.5.0
```

To generate the AirInv project documentation, the commands are:

```
cd ~/dev/sim/airinvgit
cd build
make doc
```

The AirInv project documentation is available in the following formats: HTML, LaTeX. Those documents are available in a subdirectory:

```
cd ~/dev/sim/airinvgit
cd build
cd doc
```

#### 12.11.4 Input file of AirInv Project

The schedule input file structure should look like the following sample:

Each line, beyond the header, represents a schedule entry, i.e., the specification of a given flight-period (see [AIRINV::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.

### 12.11.5 The schedule BOM Tree

The schedule-related Business Object Model (BOM) tree is a structure allowing to store all the `AIRINV::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 AirInv 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.

**12.11.5.1 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 `AIRINV::AIRINV_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 `AIRINV::AIRINV_Service::buildSampleBom()` method:

```
void buildSampleBom();
```

or can be constructed using the schedule input file described above thanks to the `AIRINV::AIRINV_Service::parseAndLoad` (`const stdair::Filename_T&`) method:

```
void parseAndLoad (const AIRINV::InventoryFilePath&);
```

### 12.11.5.2 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::airinv.cpp` program:

When the default BOM tree is used (`-b/-builtin` option of the main program `airinv.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, 0, 0, 2, 298, 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-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, 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, 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, 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, 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
```

```
SQL1 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,
SQL1 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,
SQL1 2010-Jan-20, SIN-BKK 2010-Jan-20, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 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,
SQL1 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,
SQL1 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: SQL1, 2010-Jan-21
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 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,
SQL1 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,
SQL1 2010-Jan-21, SIN-BKK 2010-Jan-21, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 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,
SQL1 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,
SQL1 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: SQL1, 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, 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
SQL1 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,
SQL1 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,
SQL1 2010-Jan-30, SIN-BKK 2010-Jan-30, Y, 1, 0, 0, 0, 0, 0, 300, 0,
SQL1 2010-Jan-30, SIN-BKK 2010-Jan-30, 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,
SQL1 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,
SQL1 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: SQL1, 2010-Jan-31
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 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, 0, 300, 0,
SQL1 2010-Jan-31, SIN-BKK 2010-Jan-31, 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,
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: SQ11, 2010-Feb-01

\*\*\*\*\*  
\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ11 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,  
SQ11 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,  
SQ11 2010-Feb-01, SIN-BKK 2010-Feb-01, Y, 1, 0, 0, 0, 0, 300, 0,  
SQ11 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,  
SQ11 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,  
SQ11 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: SQ11, 2010-Feb-02

\*\*\*\*\*  
\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ11 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, 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, 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, 0, 300, 0,
SQ11 2010-Feb-05, SIN-BKK 2010-Feb-05, 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-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, 0, 300, 0,
SQ11 2010-Feb-06, SIN-BKK 2010-Feb-06, 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-06, SIN-BKK 2010-Feb-06, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
```

```
SQL1 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: SQL1, 2010-Feb-07
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 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,
SQL1 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,
SQL1 2010-Feb-07, SIN-BKK 2010-Feb-07, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 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,
SQL1 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,
SQL1 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: SQL1, 2010-Feb-08
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 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,
SQL1 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,
SQL1 2010-Feb-08, SIN-BKK 2010-Feb-08, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 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,
```

```
SQL1 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,
SQL1 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: SQL1, 2010-Feb-09
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 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,
SQL1 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,
SQL1 2010-Feb-09, SIN-BKK 2010-Feb-09, Y, 1, 0, 0, 0, 0, 0, 300, 0,
SQL1 2010-Feb-09, SIN-BKK 2010-Feb-09, 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,
SQL1 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,
SQL1 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: SQL1, 2010-Feb-10
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 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,
SQL1 2010-Feb-10, SIN-BKK 2010-Feb-10, 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-10, SIN-BKK 2010-Feb-10, Y, 1, 0, 0, 0, 0, 0, 300, 0,
SQL1 2010-Feb-10, SIN-BKK 2010-Feb-10, 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-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, 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
SQ11 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,
SQ11 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,
SQ11 2010-Feb-12, SIN-BKK 2010-Feb-12, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 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,  
SQ11 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,  
SQ11 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: SQ11, 2010-Feb-13

\*\*\*\*\*

\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ11 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,  
SQ11 2010-Feb-13, SIN-BKK 2010-Feb-13, 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-13, SIN-BKK 2010-Feb-13, Y, 1, 0, 0, 0, 0, 300, 0,

SQ11 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,  
SQ11 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,  
SQ11 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: SQ11, 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, 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, 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, 0, 300, 0,

SQ11 2010-Feb-15, SIN-BKK 2010-Feb-15, 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-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, 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, 0, 300, 0,

SQ11 2010-Feb-16, SIN-BKK 2010-Feb-16, 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-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, 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,
```

```
SQL1 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,
SQL1 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,
SQL1 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: SQL1, 2010-Feb-21
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 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,
SQL1 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,
SQL1 2010-Feb-21, SIN-BKK 2010-Feb-21, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 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,
SQL1 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,
SQL1 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: SQL1, 2010-Feb-22
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 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,
SQL1 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,
```

```
SQL1 2010-Feb-22, SIN-BKK 2010-Feb-22, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 2010-Feb-22, SIN-BKK 2010-Feb-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,
SQL1 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,
SQL1 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: SQL1, 2010-Feb-23
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 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,
SQL1 2010-Feb-23, SIN-BKK 2010-Feb-23, 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,
SQL1 2010-Feb-23, SIN-BKK 2010-Feb-23, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 2010-Feb-23, SIN-BKK 2010-Feb-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,
SQL1 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,
SQL1 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: SQL1, 2010-Feb-24
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 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,
SQL1 2010-Feb-24, SIN-BKK 2010-Feb-24, 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-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, 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, 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, 300, 0,  
SQ11 2010-Feb-28, SIN-BKK 2010-Feb-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-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, 200, 0,  
SQ12 2010-Jan-15, SIN-HND 2010-Jan-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-Jan-15, SIN-HND 2010-Jan-15, Y, 1, Y13856, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,  
SQ12 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: SQ12, 2010-Jan-16

\*\*\*\*\*

\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ12 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,  
SQ12 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,  
SQ12 2010-Jan-16, SIN-HND 2010-Jan-16, Y, 1, 0, 0, 0, 0, 200, 0,  
SQ12 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,  
SQ12 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,  
SQ12 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: SQ12, 2010-Jan-17  
\*\*\*\*\*

\*\*\*\*\*  
\*\*\*\*\*  
Leg-Dates:  
-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ12 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,  
SQ12 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,  
SQ12 2010-Jan-17, SIN-HND 2010-Jan-17, Y, 1, 0, 0, 0, 0, 200, 0,  
SQ12 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,  
SQ12 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,  
SQ12 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: SQ12, 2010-Jan-18  
\*\*\*\*\*

\*\*\*\*\*  
\*\*\*\*\*  
Leg-Dates:  
-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ12 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,  
SQ12 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,
SQ12 2010-Jan-18, SIN-HND 2010-Jan-18, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 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,
SQ12 2010-Jan-18, SIN-HND 2010-Jan-18, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0,
SQ12 2010-Jan-18, SIN-HND 2010-Jan-18, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Jan-19
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 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,
SQ12 2010-Jan-19, SIN-HND 2010-Jan-19, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 200, 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,
SQ12 2010-Jan-19, SIN-HND 2010-Jan-19, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 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,
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,
*****
*****
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, 200, 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,
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,
SQ12 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,
SQ12 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,
SQ12 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: SQ12, 2010-Jan-30
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 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,
SQ12 2010-Feb-01, SIN-HND 2010-Feb-01, 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-01, SIN-HND 2010-Feb-01, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 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,
SQ12 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,
SQ12 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: SQ12, 2010-Feb-02
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 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,
SQ12 2010-Feb-02, SIN-HND 2010-Feb-02, 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-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, 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,  
SQ12 2010-Feb-06, SIN-HND 2010-Feb-06, Y, 1, 0, 0, 0, 0, 200, 0,  
SQ12 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,  
SQ12 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,  
SQ12 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: SQ12, 2010-Feb-07

\*\*\*\*\*  
\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity  
SQ12 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,
SQ12 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,
SQ12 2010-Feb-07, SIN-HND 2010-Feb-07, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 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,
SQ12 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,
SQ12 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: SQ12, 2010-Feb-08
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 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,
SQ12 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, 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, 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
```

```
SQL2 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,
SQL2 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,
SQL2 2010-Feb-11, SIN-HND 2010-Feb-11, Y, 1, 0, 0, 0, 0, 200, 0,
SQL2 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,
SQL2 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,
SQL2 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: SQL2, 2010-Feb-12
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 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,
SQL2 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,
SQL2 2010-Feb-12, SIN-HND 2010-Feb-12, Y, 1, 0, 0, 0, 0, 200, 0,
SQL2 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,
SQL2 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,
SQL2 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: SQL2, 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, 0, 200, 0,

SQ12 2010-Feb-21, SIN-HND 2010-Feb-21, 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-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,

SQ12 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,

SQ12 2010-Feb-22, SIN-HND 2010-Feb-22, Y, 1, 0, 0, 0, 0, 0, 200, 0,

SQ12 2010-Feb-22, SIN-HND 2010-Feb-22, 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-Feb-22, SIN-HND 2010-Feb-22, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

SQ12 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: SQ12, 2010-Feb-23

\*\*\*\*\*  
\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity

SQ12 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,

SQ12 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,

SQ12 2010-Feb-23, SIN-HND 2010-Feb-23, Y, 1, 0, 0, 0, 0, 200, 0,

SQ12 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,

SQ12 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,

SQ12 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: SQ12, 2010-Feb-24

\*\*\*\*\*  
\*\*\*\*\*

Leg-Dates:

-----

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity

SQ12 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, 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, 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,
SQ12 2010-Feb-28, SIN-HND 2010-Feb-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-Feb-28, SIN-HND 2010-Feb-28, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 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,
SQ12 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,
```

```

SQ12 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,
*****

```

### 12.11.6 Exploring the Predefined BOM Tree

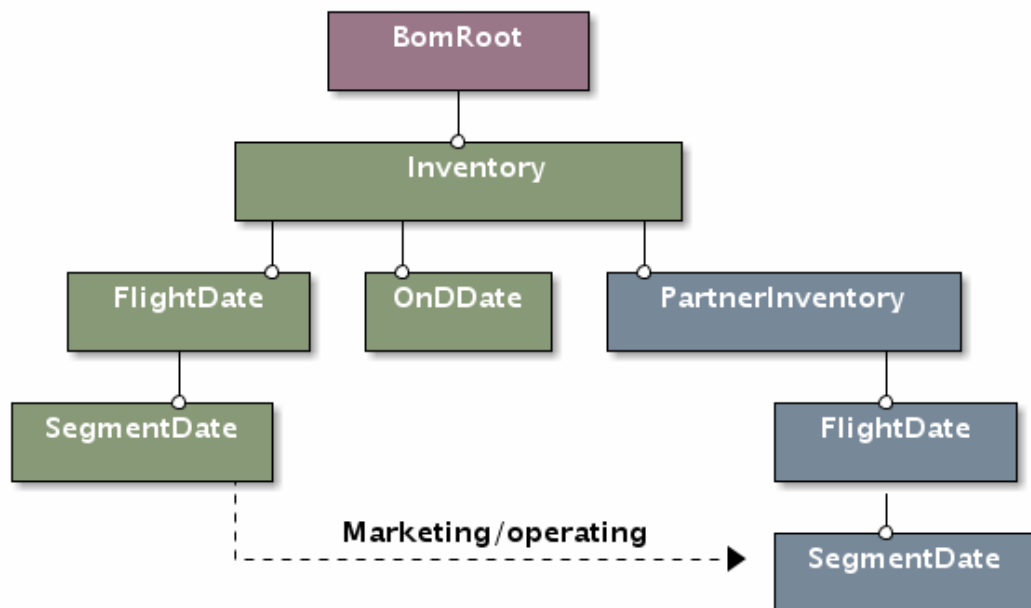


Figure 2: AirInv BOM tree

AirInv predefines a BOM (Business Object Model) tree specific to the airline IT arena.

#### 12.11.6.1 Airline Network BOM Tree

- AIRINV::ReachableUniverse
- AIRINV::OriginDestinationSet
- AIRINV::SegmentPathPeriod

#### 12.11.6.2 Airline Schedule BOM Tree

- stdair::Inventory
- stdair::FlightPeriod
- stdair::SegmentPeriod
- stdair::OnDPeriod

### 12.11.7 Extending the BOM Tree

### 12.11.8 The travel solution calculation procedure

The project AirInv aims at calculating a list of travel solutions for every incoming booking request.

## 12.12 Supported Systems

### 12.12.1 Table of Contents

- [Introduction](#)
- [.1 AirInv 0.1.x.1](#)
  - [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 AirInv 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 AirInv External](#)
    - \* [Microsoft Windows XP with MS Visual C++ and Intel MKL](#)
  - [Unix Systems](#)
    - \* [SunOS 5.9 with AirInv External](#)
- [AirInv 3.9.1](#)
- [AirInv 3.9.0](#)
- [AirInv 3.8.1](#)

### 12.12.2 Introduction

This page is intended to provide a list of AirInv supported systems, i.e. the systems on which configuration, installation and testing process of the AirInv 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 AirInv library on a system not mentioned below, please let us know, so we could update this database.

## 12.13 AirInv Supported Systems (Previous Releases)

- **Platform:** Intel Pentium 4
- **Operating System:** Fedora Core 4 (x86)
- **Compiler:** g++ (GCC) 4.0.2 20051125
- **AirInv release:** 0.1.0
- **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:** AirInv configured with:

```
% CXXFLAGS="-O3 -pipe -march=pentium4" ./configure
```

- **Date:** March 7, 2006
- **Tester:** Tony Ottosson

- **Platform:** AMD Sempron 3000+
- **Operating System:** Gentoo Linux 2006.0 (x86 arch)
- **Compiler(s):** g++ (GCC) 3.4.5
- **AirInv release:** 0.1.1
- **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
```

AirInv configured with:

```
% export CPPFLAGS="-I/usr/include/acml"
% ./configure --with-blas="-lblas"
```

- **Date:** March 31, 2006
- **Tester:** Adam Piatyszek (ediap)
- **Platform:** Intel Pentium M Centrino

- **Operating System:** Gentoo Linux 2006.0 (x86)
- **Compiler:** g++ (GCC) 3.4.5
- **AirInv release:** 0.1.1
- **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
```

AirInv configured with:

```
% ./configure --with-blas="-lblas"
```

- **Date:** March 31, 2006
- **Tester:** Adam Piatyszek (ediap)
- **Platform:** Intel Pentium M Centrino
- **Operating System:** Gentoo Linux 2006.0 (x86 arch)
- **Compiler:** g++ (GCC) 3.4.5
- **AirInv release:** 0.1.0
- **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:** AirInv 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)
- **Platform:** Intel Pentium M Centrino
- **Operating System:** Gentoo Linux 2006.0 (x86)
- **Compiler:** g++ (GCC) 3.4.5
- **AirInv release:** 0.1.1
- **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
```

AirInv configured with:

```
% ./configure --with-blas="-lblas"
```

- **Date:** March 31, 2006
- **Tester:** Adam Piatyszek (ediap)
- **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)
- **AirInv release:** 0.1.0
- **External Libraries:** BLAS, CBLAS, LAPACK and FFTW libraries from AirInv External 2.1.1 package
- **Tests Status:** All tests PASSED
- **Date:** March 7, 2006
- **Tester:** Erik G. Larsson
- **Platform:** Intel Pentium 4 CPU 3.20GHz (64-bit)
- **Operating System:** SUSE Linux 10.0 (x86\_64)
- **Compiler(s):** g++ (GCC) 4.0.2
- **AirInv release:** 0.1.0
- **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:** AirInv 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)
- **Platform:** Intel Pentium 4 CPU 3.20GHz (64-bit)
- **Operating System:** SUSE Linux 10.0 (x86\_64)
- **Compiler(s):** g++ (GCC) 4.0.2
- **AirInv release:** 0.1.0
- **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:** AirInv 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)
- **Platform:** AMD Sempron 3000+
- **Operating System:** Microsoft Windows XP SP2, Cygwin 1.5.19-4
- **Compiler(s):** g++ (GCC) 3.4.4 (cygming special)
- **AirInv release:** 0.1.1
- **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. AirInv configured with:

```
% ./configure
```

- **Date:** March 31, 2006
- **Tester:** Adam Piatyszek (ediap)
- **Platform:** AMD Sempron 3000+
- **Operating System:** Microsoft Windows XP SP2, Cygwin 1.5.19-4
- **Compiler(s):** g++ (GCC) 3.4.4 (cygming special)

- **AirInv release:** 0.1.1
- **External Libraries:** Installed from Cygwin's repository:

- `fftw-3.0.1-2`
- `fftw-dev-3.0.1-1`

ATLAS BLAS and LAPACK libraries from AirInv External 2.1.1 package configured using:

```
% ./configure --enable-atlas --disable-fftw
```

- **Tests Status:** All tests PASSED
- **Comments:** Only st

### 12.13.1 AirInv 3.9.1

### 12.13.2 AirInv 3.9.0

### 12.13.3 AirInv 3.8.1

## 12.14 Tutorials

### 12.14.1 Table of Contents

- [Preparing the AirSched 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](#)

### 12.14.2 Preparing the AirSched Project for Development

The source code for these examples can be found in the `batches` and `test/airsched` directories. They are compiled along with the rest of the `AirSched` project. See the [Users Guide](#) for more details on how to build the `AirSched` project.

### 12.14.3 Your first networkBuilde

**12.14.3.1 Summary of the different steps** All the steps below can be found in the same order in the batch `AirSched.cpp` program.

First, we instantiate the `AIRSCHEM_Service` object:

Then, we construct a default sample list of travel solutions and a default booking request (as mentioned in `ug_procedure_bookingrequest` and `ug_procedure_travelsolution` parts):

For basic use, the default BOM tree can be built using:

The main step is the network building (see [The travel solution calculation procedure](#)):

**12.14.3.2 Result of the Batch Program** When the `AirSched.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.

#### 12.14.4 Network building with an input file

**12.14.4.1 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 :

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:

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:

**12.14.4.2 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 :

**12.14.4.3 Result of the Batch Program** When the `AirSched.cpp` program is run with the `-f` option linking with the file built just above:

```
~/AirSched -f ~/<YourFileName>.csv
```

the last lines of the log output should look like:

```
[D]~/AirSchedgit/AirSched/batches/AirSched.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.

## 12.15 Command-Line Test to Demonstrate How To Test the AirInv 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>
// StdAir
#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>
#include <stdair/stdair_exceptions.hpp>
// Airinv
#include <airinv/AIRINV_Types.hpp>
#include <airinv/AIRINV_Master_Service.hpp>
#include <airinv/config/airinv-paths.hpp>

namespace boost_utf = boost::unit_test;

// (Boost) Unit Test XML Report
std::ofstream utfReportStream ("InventoryTestSuite_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() {
    }
};

// //////////////////////////////////////
bool testInventoryHelper (const unsigned short iTestFlag,
                        const stdair::Filename_T& iInventoryInputFilename,
                        const stdair::Filename_T& iScheduleInputFilename,
                        const stdair::Filename_T& iODInputFilename,
                        const stdair::Filename_T& iFRAT5InputFilename,
                        const stdair::Filename_T& iFFDisutilityInputFilename,
                        const stdair::Filename_T& iYieldInputFilename,
                        const bool isBuiltin,
                        const bool isForSchedule) {

    // Output log File
    std::ostringstream oStr;
    oStr << "InventoryTestSuite_" << iTestFlag << ".log";
    const stdair::Filename_T lLogFilename (oStr.str());

    // Set the log parameters
    std::ofstream logOutputFile;
    // Open and clean the log outputfile
    logOutputFile.open (lLogFilename.c_str());
    logOutputFile.clear();

```

```

// Initialise the AirInv service object
stdair::BasLogParams lLogParams (stdair::LOG::DEBUG,
                                logOutputFile);

// Initialise the inventory service
AIRINV::AIRINV_Master_Service airinvService (lLogParams);

// Parameters for the sale
std::string lSegmentDateKey;
stdair::ClassCode_T lClassCode;
const stdair::PartySize_T lPartySize (2);

// Check whether or not a (CSV) input file should be read
if (isBuiltin == true) {

    // Build the default sample BOM tree (filled with inventories) for AirInv
    airinvService.buildSampleBom();

    // Define a specific segment-date key for the sample BOM tree
    lSegmentDateKey = "BA,9,2011-06-10,LHR,SYD";
    lClassCode = "Q";

} else {

    if (isForSchedule == true) {
        // Build the BOM tree from parsing a schedule file (and O&D list)
        stdair::ScheduleFilePath lScheduleFilePath (iScheduleInputFilename);
        stdair::ODFilePath lODFilePath (iODInputFilename);
        stdair::FRAT5FilePath lFRAT5FilePath (iFRAT5InputFilename);
        stdair::FFDisutilityFilePath lFFDisutilityFilePath (iFFDisutilityInputFilename);
        AIRRAC::YieldFilePath lYieldFilePath (iYieldInputFilename);
        airinvService.parseAndLoad (lScheduleFilePath, lODFilePath,
                                    lFRAT5FilePath, lFFDisutilityFilePath,
                                    lYieldFilePath);

        // Define a specific segment-date key for the schedule-based inventory
        lSegmentDateKey = "SQ,11,2010-01-15,SIN,BKK";
        lClassCode = "Y";

    } else {

        // Build the BOM tree from parsing an inventory dump file
        AIRINV::InventoryFilePath lInventoryFilePath (iInventoryInputFilename);
        airinvService.parseAndLoad (lInventoryFilePath);

        // Define a specific segment-date key for the inventory parsed file
        //const std::string lSegmentDateKey ("SV, 5, 2010-03-11, KBP, JFK, 08:00:00");
        lSegmentDateKey = "SV, 5, 2010-03-11, KBP, JFK, 08:00:00";
        lClassCode = "J";

    }

}

// Make a booking
const bool hasSaleBeenSuccessful =
    airinvService.sell (lSegmentDateKey, lClassCode, lPartySize);

// DEBUG: Display the list of travel solutions
const std::string& lCSVDump = airinvService.csvDisplay();
STDAIR_LOG_DEBUG (lCSVDump);

// Close the log file
logOutputFile.close();

if (hasSaleBeenSuccessful == false) {
    STDAIR_LOG_DEBUG ("No sale can be made for '" << lSegmentDateKey
                    << "'");
}

```

```

    }

    return hasSaleBeenSuccessful;
}

// ////////////////////////////////// Main: Unit Test Suite //////////////////////////////////

// Set the UTF configuration (re-direct the output to a specific file)
BOOST_GLOBAL_FIXTURE (UnitTestConfig);

// Start the test suite
BOOST_AUTO_TEST_SUITE (master_test_suite)

BOOST_AUTO_TEST_CASE (airinv_simple_inventory_sell) {

    // Input file name
    const stdair::Filename_T lInventoryInputFilename (STDAIR_SAMPLE_DIR
                                                    "/invdump01.csv");

    // State whether the BOM tree should be built-in or parsed from an input file
    const bool isBuiltin = false;
    // State whether the BOM tree should be built from a schedule file (instead of from an inventory dump)
    const bool isForSchedule = false;

    // Try sell a default segment.
    bool hasTestBeenSuccessful = false;
    BOOST_CHECK_NO_THROW (hasTestBeenSuccessful =
                          testInventoryHelper (0, lInventoryInputFilename,
                                              " ", " ", " ", " ", " ", isBuiltin, isForSchedule));
    BOOST_CHECK_EQUAL (hasTestBeenSuccessful, true);
}

BOOST_AUTO_TEST_CASE (airinv_simple_inventory_sell_built_in) {

    // State whether the BOM tree should be built-in or parsed from an input file
    const bool isBuiltin = true;
    // State whether the BOM tree should be built from a schedule file (instead of from an inventory dump)
    const bool isForSchedule = false;

    // Try sell a default segment.
    bool hasTestBeenSuccessful = false;
    BOOST_CHECK_NO_THROW (hasTestBeenSuccessful =
                          testInventoryHelper (1, " ", " ", " ", " ", " ", " ", " ",
                                              isBuiltin, isForSchedule));
    BOOST_CHECK_EQUAL (hasTestBeenSuccessful, true);
}

BOOST_AUTO_TEST_CASE (airinv_simple_inventory_sell_schedule) {

    // Input file names
    const stdair::Filename_T lScheduleInputFilename (STDAIR_SAMPLE_DIR
                                                    "/schedule01.csv");
    const stdair::Filename_T lODInputFilename (STDAIR_SAMPLE_DIR
                                              "/ond01.csv");
    const stdair::Filename_T lFRAT5InputFilename (STDAIR_SAMPLE_DIR
                                                  "/frat5.csv");
    const stdair::Filename_T lFFDisutilityInputFilename (STDAIR_SAMPLE_DIR
                                                         "/ffDisutility.csv");
    const stdair::Filename_T lYieldInputFilename (STDAIR_SAMPLE_DIR
                                                  "/yieldstore01.csv");

    // State whether the BOM tree should be built-in or parsed from an input file
    const bool isBuiltin = false;

```



```

// State whether the BOM tree should be built from a schedule file (instead of from an inventory dump)
const bool isForSchedule = true;

// Try sell a default segment.
bool hasTestBeenSuccessful = false;
BOOST_CHECK_NO_THROW (hasTestBeenSuccessful =
    testInventoryHelper (2, " ",
                        lScheduleInputFilename,
                        lODInputFilename,
                        lFRAT5InputFilename,
                        lFFDisutilityInputFilename,
                        lYieldInputFilename,
                        isBuiltin, isForSchedule));

BOOST_CHECK_EQUAL (hasTestBeenSuccessful, true);
}

BOOST_AUTO_TEST_CASE (airinv_error_inventory_input_file) {

    // Inventory input file name
    const stdair::Filename_T lMissingInventoryFilename (STDAIR_SAMPLE_DIR
                                                         "/missingFile.csv");

    // State whether the BOM tree should be built-in or parsed from an input file
    const bool isBuiltin = false;
    // State whether the BOM tree should be built from a schedule file (instead of from an inventory dump)
    const bool isForSchedule = false;

    // Try sell a default segment.
    BOOST_CHECK_THROW (testInventoryHelper (3, lMissingInventoryFilename,
                                           " ", " ", " ", " ", " ", isBuiltin, isForSchedule),
                      AIRINV::InventoryInputFileNotFoundException);
}

BOOST_AUTO_TEST_CASE (airinv_error_schedule_input_file) {

    // Schedule input file name
    const stdair::Filename_T lMissingScheduleFilename (STDAIR_SAMPLE_DIR
                                                         "/missingFile.csv");
    const stdair::Filename_T lFRAT5InputFilename (STDAIR_SAMPLE_DIR
                                                    "/frat5.csv");
    const stdair::Filename_T lFFDisutilityInputFilename (STDAIR_SAMPLE_DIR
                                                          "/ffDisutility.csv");

    // State whether the BOM tree should be built-in or parsed from an input file
    const bool isBuiltin = false;
    // State whether the BOM tree should be built from a schedule file (instead of from an inventory dump)
    const bool isForSchedule = true;

    // Try sell a default segment.
    BOOST_CHECK_THROW (testInventoryHelper (4, " ", lMissingScheduleFilename,
                                           " ", lFRAT5InputFilename,
                                           lFFDisutilityInputFilename, " ",
                                           isBuiltin, isForSchedule),
                      AIRINV::ScheduleInputFileNotFoundException);
}

BOOST_AUTO_TEST_CASE (airinv_error_yield_input_file) {

    // Input file names
    const stdair::Filename_T lScheduleInputFilename (STDAIR_SAMPLE_DIR
                                                         "/schedule01.csv");
    const stdair::Filename_T lODInputFilename (STDAIR_SAMPLE_DIR
                                                  "/ond01.csv");
    const stdair::Filename_T lFRAT5InputFilename (STDAIR_SAMPLE_DIR

```

```

        "/frat5.csv");
const stdair::Filename_T lFFDisutilityInputFilename (STDAIR_SAMPLE_DIR
        "/ffDisutility.csv");
const stdair::Filename_T lYieldInputFilename (STDAIR_SAMPLE_DIR
        "/missingFile.csv");

// State whether the BOM tree should be built-in or parsed from an input file
const bool isBuiltin = false;
// State whether the BOM tree should be built from a schedule file (instead of from an inventory dump)
const bool isForSchedule = true;

// Try sell a default segment.
BOOST_CHECK_THROW (testInventoryHelper (5, " ",
        lScheduleInputFilename,
        lODInputFilename,
        lFRAT5InputFilename,
        lFFDisutilityInputFilename,
        lYieldInputFilename,
        isBuiltin, isForSchedule),
        AIRRAC::YieldInputFileNotFoundException);
}

BOOST_AUTO_TEST_CASE (airinv_error_flight_date_duplication) {
    // Input file names
    const stdair::Filename_T lScheduleInputFilename (STDAIR_SAMPLE_DIR
        "/scheduleError01.csv");
    const stdair::Filename_T lODInputFilename (STDAIR_SAMPLE_DIR
        "/ond01.csv");
    const stdair::Filename_T lFRAT5InputFilename (STDAIR_SAMPLE_DIR
        "/frat5.csv");
    const stdair::Filename_T lFFDisutilityInputFilename (STDAIR_SAMPLE_DIR
        "/ffDisutility.csv");
    const stdair::Filename_T lYieldInputFilename (STDAIR_SAMPLE_DIR
        "/missingFile.csv");

    // State whether the BOM tree should be built-in or parsed from an input file
    const bool isBuiltin = false;
    // State whether the BOM tree should be built from a schedule file (instead of from an inventory dump)
    const bool isForSchedule = true;

    // Try sell a default segment.
    BOOST_CHECK_THROW (testInventoryHelper (6, " ",
        lScheduleInputFilename,
        lODInputFilename,
        lFRAT5InputFilename,
        lFFDisutilityInputFilename,
        lYieldInputFilename,
        isBuiltin, isForSchedule),
        AIRINV::FlightDateDuplicationException);
}

BOOST_AUTO_TEST_CASE (airinv_error_schedule_parsing_failed) {
    // Input file names
    const stdair::Filename_T lScheduleInputFilename (STDAIR_SAMPLE_DIR
        "/scheduleError02.csv");
    const stdair::Filename_T lODInputFilename (STDAIR_SAMPLE_DIR
        "/ond01.csv");
    const stdair::Filename_T lFRAT5InputFilename (STDAIR_SAMPLE_DIR
        "/frat5.csv");
    const stdair::Filename_T lFFDisutilityInputFilename (STDAIR_SAMPLE_DIR
        "/ffDisutility.csv");
    const stdair::Filename_T lYieldInputFilename (STDAIR_SAMPLE_DIR
        "/yieldstore01.csv");

```

```
// State whether the BOM tree should be built-in or parsed from an input file
const bool isBuiltin = false;
// State whether the BOM tree should be built from a schedule file (instead of from an inventory dump)
const bool isForSchedule = true;

// Try sell a default segment.
BOOST_CHECK_THROW (testInventoryHelper (7, " ",
                                         lScheduleInputFilename,
                                         lODInputFilename,
                                         lFRAT5InputFilename,
                                         lFFDisutilityInputFilename,
                                         lYieldInputFilename,
                                         isBuiltin, isForSchedule),
                  AIRINV::ScheduleFileParsingFailedException);

}

// End the test suite
BOOST_AUTO_TEST_SUITE_END()

/*!
```