

SimCRS

0.1.1

Generated by Doxygen 1.8.1.2

Wed Aug 29 2012 20:26:01

## Contents

<b>1</b>	<b>SimCRS Documentation</b>	<b>1</b>
1.1	Getting Started	1
1.2	SimCRS at SourceForge	1
1.3	SimCRS Development	1
1.4	External Libraries	1
1.5	Support SimCRS	2
1.6	About SimCRS	2
<b>2</b>	<b>People</b>	<b>2</b>
2.1	Project Admins	2
2.2	Developers	2
2.3	Retired Developers	2
2.4	Contributors	2
2.5	Distribution Maintainers	3
<b>3</b>	<b>Coding Rules</b>	<b>3</b>
3.1	Default Naming Rules for Variables	3
3.2	Default Naming Rules for Functions	3
3.3	Default Naming Rules for Classes and Structures	3
3.4	Default Naming Rules for Files	3
3.5	Default Functionality of Classes	3
<b>4</b>	<b>Copyright and License</b>	<b>4</b>
4.1	GNU LESSER GENERAL PUBLIC LICENSE	4
4.1.1	Version 2.1, February 1999	4
4.2	Preamble	4
4.3	TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION	5
4.3.1	NO WARRANTY	9
4.3.2	END OF TERMS AND CONDITIONS	9
4.4	How to Apply These Terms to Your New Programs	9
<b>5</b>	<b>Documentation Rules</b>	<b>10</b>
5.1	General Rules	10
5.2	File Header	11
5.3	Grouping Various Parts	11
<b>6</b>	<b>Main features</b>	<b>12</b>
6.1	Network generation	12
6.2	Inventory generation	12
6.3	Finding travel solutions	12

6.4	Distributed inventories . . . . .	12
6.5	Other features . . . . .	12
<b>7</b>	<b>Make a Difference</b>	<b>12</b>
<b>8</b>	<b>Make a new release</b>	<b>13</b>
8.1	Introduction . . . . .	13
8.2	Initialisation . . . . .	13
8.3	Release branch maintenance . . . . .	13
8.4	Commit and publish the release branch . . . . .	13
8.5	Create distribution packages . . . . .	13
8.6	Upload the HTML documentation to SourceForge . . . . .	14
8.7	Generate the RPM packages . . . . .	14
8.8	Update distributed change log . . . . .	14
8.9	Create the binary package, including the documentation . . . . .	15
8.10	Upload the files to SourceForge . . . . .	15
8.11	Make a new post . . . . .	15
8.12	Send an email on the announcement mailing-list . . . . .	15
<b>9</b>	<b>Installation</b>	<b>15</b>
9.1	Table of Contents . . . . .	15
9.2	Fedora/RedHat Linux distributions . . . . .	16
9.3	SimCRS Requirements . . . . .	16
9.4	Basic Installation . . . . .	16
9.5	Compilers and Options . . . . .	17
9.6	Compiling For Multiple Architectures . . . . .	17
9.7	Installation Names . . . . .	18
9.8	Optional Features . . . . .	19
9.9	Particular systems . . . . .	19
9.10	Specifying the System Type . . . . .	20
9.11	Sharing Defaults . . . . .	20
9.12	Defining Variables . . . . .	20
9.13	'cmake' Invocation . . . . .	20
<b>10</b>	<b>Linking with SimCRS</b>	<b>25</b>
10.1	Table of Contents . . . . .	25
10.2	Introduction . . . . .	25
10.3	Dependencies . . . . .	25
10.3.1	StdAir . . . . .	25
10.3.2	Other Simulation-Related Components . . . . .	25
10.4	Using the pkg-config command . . . . .	26

10.5 Using the simcrs-config script . . . . .	27
10.6 M4 macro for the GNU Autotools . . . . .	27
10.7 Using SimCRS with dynamic linking . . . . .	27
<b>11 Test Rules</b>	<b>27</b>
11.1 The Test Source Files . . . . .	27
11.2 The Reference File . . . . .	28
11.3 Testing SimCRS Library . . . . .	28
<b>12 Users Guide</b>	<b>28</b>
12.1 Table of Contents . . . . .	28
12.2 Introduction . . . . .	28
12.3 Get Started . . . . .	29
12.3.1 Get the SimCRS library . . . . .	29
12.3.2 Build the SimCRS project . . . . .	29
12.3.3 Build and Run the Tests . . . . .	29
12.3.4 Install the SimCRS Project (Binaries, Documentation) . . . . .	29
12.4 Input file of SimCRS Project . . . . .	30
12.5 The schedule BOM Tree . . . . .	31
12.5.1 Build of the schedule BOM tree . . . . .	31
12.5.2 Display of the schedule BOM tree . . . . .	31
12.6 Exploring the Predefined BOM Tree . . . . .	75
12.6.1 Airline Network BOM Tree . . . . .	75
12.6.2 Airline Schedule BOM Tree . . . . .	75
12.7 Extending the BOM Tree . . . . .	75
12.8 The travel solution calculation procedure . . . . .	75
<b>13 Supported Systems</b>	<b>76</b>
13.1 Table of Contents . . . . .	76
13.2 Introduction . . . . .	76
<b>14 SimCRS Supported Systems (Previous Releases)</b>	<b>76</b>
14.1 SimCRS 3.9.1 . . . . .	76
14.2 SimCRS 3.9.0 . . . . .	76
14.3 SimCRS 3.8.1 . . . . .	76
<b>15 Tutorials</b>	<b>77</b>
15.1 Table of Contents . . . . .	77
15.2 Preparing the AirSched Project for Development . . . . .	77
15.3 Your first networkBuilde . . . . .	77
15.3.1 Summary of the different steps . . . . .	77
15.3.2 Result of the Batch Program . . . . .	77

15.4	Network building with an input file . . . . .	78
15.4.1	How to build a network input file? . . . . .	78
15.4.2	Building the BOM tree with an input file . . . . .	79
15.4.3	Result of the Batch Program . . . . .	79
<b>16</b>	<b>Command-Line Test to Demonstrate How To Test the SimCRS Project</b>	<b>79</b>
<b>17</b>	<b>Namespace Index</b>	<b>82</b>
17.1	Namespace List . . . . .	82
<b>18</b>	<b>Class Index</b>	<b>82</b>
18.1	Class Hierarchy . . . . .	82
<b>19</b>	<b>Class Index</b>	<b>83</b>
19.1	Class List . . . . .	83
<b>20</b>	<b>File Index</b>	<b>83</b>
20.1	File List . . . . .	83
<b>21</b>	<b>Namespace Documentation</b>	<b>84</b>
21.1	AIRINV Namespace Reference . . . . .	84
21.2	SIMCRS Namespace Reference . . . . .	84
21.2.1	Typedef Documentation . . . . .	85
21.2.2	Variable Documentation . . . . .	85
21.3	stdair Namespace Reference . . . . .	85
21.3.1	Detailed Description . . . . .	85
<b>22</b>	<b>Class Documentation</b>	<b>86</b>
22.1	SIMCRS::AvailabilityRetrievalException Class Reference . . . . .	86
22.1.1	Detailed Description . . . . .	86
22.2	SIMCRS::BomAbstract Class Reference . . . . .	86
22.2.1	Detailed Description . . . . .	86
22.2.2	Constructor & Destructor Documentation . . . . .	87
22.2.3	Member Function Documentation . . . . .	87
22.2.4	Friends And Related Function Documentation . . . . .	87
22.3	SIMCRS::BookingException Class Reference . . . . .	88
22.3.1	Detailed Description . . . . .	88
22.4	SIMCRS::DistributionManager Class Reference . . . . .	88
22.4.1	Detailed Description . . . . .	88
22.4.2	Friends And Related Function Documentation . . . . .	88
22.5	SIMCRS::FacBomAbstract Class Reference . . . . .	88
22.5.1	Detailed Description . . . . .	89
22.5.2	Member Typedef Documentation . . . . .	89

22.5.3	Constructor & Destructor Documentation . . . . .	89
22.5.4	Member Function Documentation . . . . .	90
22.5.5	Friends And Related Function Documentation . . . . .	90
22.5.6	Member Data Documentation . . . . .	90
22.6	SIMCRS::FacServiceAbstract Class Reference . . . . .	91
22.6.1	Detailed Description . . . . .	91
22.6.2	Member Typedef Documentation . . . . .	91
22.6.3	Constructor & Destructor Documentation . . . . .	91
22.6.4	Member Function Documentation . . . . .	92
22.6.5	Member Data Documentation . . . . .	92
22.7	SIMCRS::FacSimcrsServiceContext Class Reference . . . . .	92
22.7.1	Detailed Description . . . . .	93
22.7.2	Member Typedef Documentation . . . . .	93
22.7.3	Constructor & Destructor Documentation . . . . .	93
22.7.4	Member Function Documentation . . . . .	93
22.7.5	Member Data Documentation . . . . .	94
22.8	SIMCRS::FacSupervisor Class Reference . . . . .	94
22.8.1	Detailed Description . . . . .	95
22.8.2	Member Typedef Documentation . . . . .	95
22.8.3	Constructor & Destructor Documentation . . . . .	95
22.8.4	Member Function Documentation . . . . .	96
22.9	RootException Class Reference . . . . .	97
22.10	SIMCRS::ServiceAbstract Class Reference . . . . .	97
22.10.1	Detailed Description . . . . .	97
22.10.2	Constructor & Destructor Documentation . . . . .	98
22.10.3	Member Function Documentation . . . . .	98
22.11	SIMCRS::SIMCRS_Service Class Reference . . . . .	98
22.11.1	Detailed Description . . . . .	99
22.11.2	Constructor & Destructor Documentation . . . . .	99
22.11.3	Member Function Documentation . . . . .	100
22.12	SIMCRS::SIMCRS_ServiceContext Class Reference . . . . .	104
22.12.1	Detailed Description . . . . .	104
22.12.2	Member Function Documentation . . . . .	104
22.12.3	Friends And Related Function Documentation . . . . .	105
<b>23</b>	<b>File Documentation</b>	<b>105</b>
23.1	doc/local/authors.doc File Reference . . . . .	105
23.2	doc/local/codingrules.doc File Reference . . . . .	105
23.3	doc/local/copyright.doc File Reference . . . . .	105
23.4	doc/local/documentation.doc File Reference . . . . .	105

23.5 doc/local/features.doc File Reference . . . . .	105
23.6 doc/local/help_wanted.doc File Reference . . . . .	105
23.7 doc/local/howto_release.doc File Reference . . . . .	105
23.8 doc/local/index.doc File Reference . . . . .	105
23.9 doc/local/installation.doc File Reference . . . . .	105
23.10 doc/local/linking.doc File Reference . . . . .	105
23.11 doc/local/test.doc File Reference . . . . .	105
23.12 doc/local/users_guide.doc File Reference . . . . .	105
23.13 doc/local/verification.doc File Reference . . . . .	105
23.14 doc/tutorial/tutorial.doc File Reference . . . . .	105
23.15 simcrs/basic/BasConst.cpp File Reference . . . . .	105
23.16 BasConst.cpp . . . . .	106
23.17 simcrs/basic/BasConst_General.hpp File Reference . . . . .	106
23.18 BasConst_General.hpp . . . . .	106
23.19 simcrs/basic/BasConst_SIMCRS_Service.hpp File Reference . . . . .	106
23.20 BasConst_SIMCRS_Service.hpp . . . . .	106
23.21 simcrs/batches/simcrs.cpp File Reference . . . . .	107
23.21.1 Function Documentation . . . . .	107
23.21.2 Variable Documentation . . . . .	109
23.22 simcrs.cpp . . . . .	109
23.23 simcrs/bom/BomAbstract.cpp File Reference . . . . .	114
23.24 BomAbstract.cpp . . . . .	114
23.25 simcrs/bom/BomAbstract.hpp File Reference . . . . .	114
23.25.1 Function Documentation . . . . .	115
23.26 BomAbstract.hpp . . . . .	115
23.27 simcrs/command/DistributionManager.cpp File Reference . . . . .	116
23.28 DistributionManager.cpp . . . . .	116
23.29 simcrs/command/DistributionManager.hpp File Reference . . . . .	117
23.30 DistributionManager.hpp . . . . .	118
23.31 simcrs/config/simcrs-paths.hpp File Reference . . . . .	118
23.31.1 Macro Definition Documentation . . . . .	119
23.32 simcrs-paths.hpp . . . . .	120
23.33 simcrs/config/simcrs-paths.hpp.in File Reference . . . . .	121
23.33.1 Macro Definition Documentation . . . . .	121
23.34 simcrs-paths.hpp.in . . . . .	122
23.35 simcrs/factory/FacBomAbstract.cpp File Reference . . . . .	123
23.36 FacBomAbstract.cpp . . . . .	123
23.37 simcrs/factory/FacBomAbstract.hpp File Reference . . . . .	124
23.38 FacBomAbstract.hpp . . . . .	124
23.39 simcrs/factory/FacServiceAbstract.cpp File Reference . . . . .	125

<a href="#">23.40FacServiceAbstract.cpp</a>	125
<a href="#">23.41simcrs/factory/FacServiceAbstract.hpp File Reference</a>	125
<a href="#">23.42FacServiceAbstract.hpp</a>	125
<a href="#">23.43simcrs/factory/FacSimcrsServiceContext.cpp File Reference</a>	126
<a href="#">23.44FacSimcrsServiceContext.cpp</a>	126
<a href="#">23.45simcrs/factory/FacSimcrsServiceContext.hpp File Reference</a>	127
<a href="#">23.46FacSimcrsServiceContext.hpp</a>	127
<a href="#">23.47simcrs/factory/FacSupervisor.cpp File Reference</a>	128
<a href="#">23.48FacSupervisor.cpp</a>	128
<a href="#">23.49simcrs/factory/FacSupervisor.hpp File Reference</a>	129
<a href="#">23.50FacSupervisor.hpp</a>	129
<a href="#">23.51simcrs/service/ServiceAbstract.cpp File Reference</a>	130
<a href="#">23.52ServiceAbstract.cpp</a>	130
<a href="#">23.53simcrs/service/ServiceAbstract.hpp File Reference</a>	130
<a href="#">23.53.1 Function Documentation</a>	131
<a href="#">23.54ServiceAbstract.hpp</a>	131
<a href="#">23.55simcrs/service/SIMCRS_Service.cpp File Reference</a>	132
<a href="#">23.56SIMCRS_Service.cpp</a>	132
<a href="#">23.57simcrs/service/SIMCRS_ServiceContext.cpp File Reference</a>	140
<a href="#">23.58SIMCRS_ServiceContext.cpp</a>	140
<a href="#">23.59simcrs/service/SIMCRS_ServiceContext.hpp File Reference</a>	141
<a href="#">23.60SIMCRS_ServiceContext.hpp</a>	142
<a href="#">23.61simcrs/SIMCRS_Service.hpp File Reference</a>	143
<a href="#">23.62SIMCRS_Service.hpp</a>	144
<a href="#">23.63simcrs/SIMCRS_Types.hpp File Reference</a>	145
<a href="#">23.64SIMCRS_Types.hpp</a>	146
<a href="#">23.65test/simcrs/CRSTestSuite.cpp File Reference</a>	146
<a href="#">23.66CRSTestSuite.cpp</a>	146

## 1 SimCRS Documentation

### 1.1 Getting Started

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



- [Make a new release](#)
- [People](#)

## 1.2 SimCRS at SourceForge

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

## 1.3 SimCRS 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 SimCRS

## 1.6 About SimCRS

SimCRS is a C++ library of travel distribution classes and functions, exclusively targeting simulation purposes. [N](#)

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

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

### Note

(N) - The SimCRS library is **NOT** intended, in any way, to be used by any entity 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 SimCRS.

## 2 People

### 2.1 Project Admins

- Denis Arnaud [denis\\_arnaud@users.sourceforge.net](mailto:denis_arnaud@users.sourceforge.net) (N)
- Anh Quan Nguyen [quannaus@users.sourceforge.net](mailto:quannaus@users.sourceforge.net) (N)

### 2.2 Developers

- Anh Quan Nguyen [quannaus@users.sourceforge.net](mailto:quannaus@users.sourceforge.net) (N)
- Denis Arnaud [denis\\_arnaud@users.sourceforge.net](mailto:denis_arnaud@users.sourceforge.net) (N)
- Son Nguyen Kim [snguyenkim@users.sourceforge.net](mailto:snguyenkim@users.sourceforge.net)
- Nicolas Bondoux [nbondoux@users.sourceforge.net](mailto:nbondoux@users.sourceforge.net) (N)

### 2.3 Retired Developers

- Patrick Grandjean [pgrandjean@users.sourceforge.net](mailto:pgrandjean@users.sourceforge.net) (N)
- Ngoc-Thach Hoang [hoangngocthach@users.sourceforge.net](mailto:hoangngocthach@users.sourceforge.net) (N)

### 2.4 Contributors

- Emmanuel Bastien [ebastien@users.sourceforge.net](mailto:ebastien@users.sourceforge.net) (N)
- Christophe Lacombe [ddtof@users.sourceforge.net](mailto:ddtof@users.sourceforge.net) (N)

### 2.5 Distribution Maintainers

- **Fedora/RedHat**: Denis Arnaud [denis\\_arnaud@users.sourceforge.net](mailto:denis_arnaud@users.sourceforge.net) (N)
- **Debian**: Emmanuel Bastien [ebastien@users.sourceforge.net](mailto:ebastien@users.sourceforge.net) (N)

#### Note

(N) - [Amadeus](#) employees.

## 3 Coding Rules

In the following sections we describe the naming conventions which are used for files, classes, structures, local variables, and global variables.

### 3.1 Default Naming Rules for Variables

Variables names follow Java naming conventions. Examples:

- `lNumberOfPassengers`
- `lSeatAvailability`

### 3.2 Default Naming Rules for Functions

Function names follow Java naming conventions. Example:

- `int myFunctionName (const int& a, int b)`

### 3.3 Default Naming Rules for Classes and Structures

Each new word in a class or structure name should always start with a capital letter and the words should be separated with an under-score. Abbreviations are written with capital letters. Examples:

- `MyClassName`
- `MyStructName`

### 3.4 Default Naming Rules for Files

Files are named after the C++ class names.

Source files are named using `.cpp` suffix, whereas header files end with `.hpp` extension. Examples:

- `FlightDate.hpp`
- `SegmentDate.cpp`

### 3.5 Default Functionality of Classes

All classes that are configured by input parameters should include:

- default empty constructor
- one or more additional constructor(s) that takes input parameters and initializes the class instance
- setup function, preferably named `'setup'` or `'set_parameters'`

Explicit destructor functions are not required, unless they are needed. It shall not be possible to use any of the other member functions unless the class has been properly initiated with the input parameters.

## 4 Copyright and License

### 4.1 GNU LESSER GENERAL PUBLIC LICENSE

#### 4.1.1 Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

## 4.2 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages—typically libraries—of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

### 4.3 TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

1. 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.

1. 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.

1. 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.

1. 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.

1. 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.

1. 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.

1. 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.

1. 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.

1. 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.
1. 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.

1. 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.
1. 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.

1. 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.

#### 4.3.1 NO WARRANTY

1. 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.



1. 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.

#### 4.3.2 END OF TERMS AND CONDITIONS

### 4.4 How to Apply These Terms to Your New Programs

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.
```

```
This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.
```

```
You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
```

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
library 'Frob' (a library for tweaking knobs) written by James Random Hacker.
```

```
<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice
```

That's all there is to it!

Source

## 5 Documentation Rules

### 5.1 General Rules

All classes in SimCRS 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 SimCRS 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
};

```

## 5.2 File Header

All files should start with the following header, which include Doxygen's \file, \brief and \author tags, \$Date\$ and \$Revisions\$ CVS tags, and a common copyright note:

```

/*!
 * \file
 * \brief Brief description of the file here
 * \author Names of the authors who contributed to this code
 * \date Date
 *
 * Detailed description of the file here if needed.
 *
 * -----
 *
 * SimCRS - C++ Simulated Travel Distribution System Library
 *
 * Copyright (C) 2009-2011 (\see authors file for a list of contributors)
 *
 * \see copyright file for license information
 */

```

```
* -----
*/
```

### 5.3 Grouping Various Parts

All functions must be added to a Doxygen group in order to appear in the documentation. The following code example defines the group `'my_group'`:

```
/*!
 * \defgroup my_group Brief description of the group here
 *
 * Detailed description of the group here
 */
```

The following example shows how to document the function `myFunction` and how to add it to the group `my_group`:

```
/*!
 * \brief Brief description of myFunction here
 * \ingroup my_group
 *
 * Detailed description of myFunction here
 *
 * \param[in] param1 Description of \a param1 here
 * \param[in] param2 Description of \a param2 here
 * \return Description of the return value here
 */
TYPE3 myFunction(TYPE1 param1, TYPE2 &param2);
```

## 6 Main features

A short list of the main features of SimCRS is given below sorted in different categories. Many more features and functions exist and for these we refer to the reference documentation.

### 6.1 Network generation

- Network/graph generation

### 6.2 Inventory generation

- Inventory generation

### 6.3 Finding travel solutions

- Matching of travel solutions with user requests

### 6.4 Distributed inventories

- Inventory independent partitions
- MPI-based distribution

## 6.5 Other features

- CSV input file parsing
- Memory handling

## 7 Make a Difference

**Do not ask what SimCRS can do for you. Ask what you can do for SimCRS.**

You can help us to develop the SimCRS library. There are always a lot of things you can do:

- Start using SimCRS
- Tell your friends about SimCRS 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 SimCRS discussion forums on SourceForge. If you know the answer to a question, help others to overcome their SimCRS 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 SimCRS to new platforms. If you manage to compile SimCRS on a new platform, then tell us how you did it.
- Send us your code. If you have a good SimCRS compatible code, which you can release under the LGPL, and you think it should be included in SimCRS, then send it to us.
- Become an SimCRS developer. Send us an e-mail and tell what you can do for SimCRS.

## 8 Make a new release

### 8.1 Introduction

This document describes briefly the recommended procedure of releasing a new version of SimCRS using a Linux development machine and the SourceForge project site.

The following steps are required to make a release of the distribution package.

### 8.2 Initialisation

Clone locally the full [Git project](#):

```
cd ~
mkdir -p dev/sim
cd ~/dev/sim
git clone git://simcrs.git.sourceforge.net/gitroot/simcrs/simcrs simcrsgit
cd simcrsgit
git checkout trunk
```

### 8.3 Release branch maintenance

Switch to the release branch, on your local clone, and merge the latest updates from the trunk. Decide about the new version to be released.

```
cd ~/dev/sim/simcrsgit
git checkout releases
git merge trunk
```

Update the version in the various build system files, replacing the old version numbers by the correct ones:

```
vi CMakeLists.txt
vi autogen.sh
vi README
```

Update the version, add some news in the NEWS file, add a change-log in the ChangeLog file and in the RPM specification files:

```
vi NEWS
vi ChangeLog
vi simcrs.spec
```

### 8.4 Commit and publish the release branch

Commit the new release:

```
cd ~/dev/sim/simcrsgit
git add -A
git commit -m "[Release 0.5.0] Release of the 0.5.0 version of SimCRS."
git push
```

### 8.5 Create distribution packages

Create the distribution packages using the following command:

```
cd ~/dev/sim/simcrsgit
git checkout releases
rm -rf build && mkdir -p build
cd build
export INSTALL_BASEDIR=/home/user/dev/deliveries
export LIBSUFFIX_4_CMAKE="-DLIB_SUFFIX=64"
cmake -DCMAKE_INSTALL_PREFIX=${INSTALL_BASEDIR}/simcrs-0.5.0 \
  -DWITH_STDAIR_PREFIX=${INSTALL_BASEDIR}/stdair-stable \
  -DWITH_AIRRAC_PREFIX=${INSTALL_BASEDIR}/airsched-stable \
  -DWITH_AIRRAC_PREFIX=${INSTALL_BASEDIR}/airrac-stable \
  -DWITH_RMOL_PREFIX=${INSTALL_BASEDIR}/rmol-stable \
  -DWITH_RMOL_PREFIX=${INSTALL_BASEDIR}/airinv-stable \
  -DWITH_RMOL_PREFIX=${INSTALL_BASEDIR}/simfqt-stable \
  -DCMAKE_BUILD_TYPE:String=Debug -DINSTALL_DOC:BOOL=ON \
  ${LIBSUFFIX_4_CMAKE} ..
make check && make dist
make install
```

This will configure, compile and check the package. The output packages will be named, for instance, `simcrs-0.5.0.tar.gz` and `simcrs-0.5.0.tar.bz2`.

### 8.6 Upload the HTML documentation to SourceForge

In order to update the Web site files, either:

- **synchronise them with rsync and SSH:** Upload the just generated HTML (and PDF) documentation onto the **SourceForge Web site**.

```
cd ~/dev/sim/simcrsgit/build
git checkout releases
rsync -aiv ${INSTALL_BASEDIR}/simcrs-0.5.0/share/doc/simcrs-0.5.0/html/ \
  your_sf_user,simcrs@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](#).

## 8.7 Generate the RPM packages

Optionally, generate the RPM package (for instance, for [Fedora/RedHat](#)):

```
cd ~/dev/sim/simcrsgit/build
git checkout releases
make dist
```

To perform this step, `rpm-build`, `rpmlint` and `rpmdevtools` have to be available on the system.

```
cp ../simcrs.spec ~/dev/packages/SPECS \
  && cp simcrs-0.5.0.tar.bz2 ~/dev/packages/SOURCES
cd ~/dev/packages/SPECS
rpmbuild -ba simcrs.spec
cd ~/dev/packages
rpmlint -i SPECS/simcrs.spec SRPMS/simcrs-0.5.0-1.fc16.src.rpm \
  RPMS/noarch/simcrs-* RPMS/i686/simcrs-*
```

## 8.8 Update distributed change log

Update the `NEWS` and `ChangeLog` files with appropriate information, including what has changed since the previous release. Then commit and push the changes into the [SimCRS's Git repository](#).

## 8.9 Create the binary package, including the documentation

Create the binary package, which includes HTML and PDF documentation, using the following command:

```
cd ~/dev/sim/simcrsgit/build
git checkout releases
make package
```

The output binary package will be named, for instance, `simcrs-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.

## 8.10 Upload the files to SourceForge

Upload the distribution and documentation packages to the SourceForge server. Check [SourceForge help page on uploading software](#).

## 8.11 Make a new post

- submit a new entry in the [SourceForge project-related news feed](#)
- make a new post on the [SourceForge hosted WordPress blog](#)
- and update, if necessary, [Trac tickets](#).

## 8.12 Send an email on the announcement mailing-list

Finally, you should send an announcement to [simcrs-announce@lists.sourceforge.net](mailto:simcrs-announce@lists.sourceforge.net) (see <https://lists.sourceforge.net/lists/listinfo/simcrs-announce> for the archives)

# 9 Installation

## 9.1 Table of Contents

- [Fedora/RedHat Linux distributions](#)
- [SimCRS 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](#)

## 9.2 Fedora/RedHat Linux distributions

Note that on [Fedora/RedHat](#) Linux distributions, RPM packages are available and can be installed with your usual package manager. For instance:

```
yum -y install simcrs-devel simcrs-doc
```

RPM packages can also be available on the [SourceForge download site](#).

## 9.3 SimCRS Requirements

SimCRS should compile without errors or warnings on most GNU/Linux systems, on UNIX systems like Solaris SunOS, and on POSIX based environments for Microsoft Windows like Cygwin or MinGW with MSYS. It can be also built on Microsoft Windows NT/2000/XP/Vista/7 using Microsoft's Visual C++ .NET, but our support for this compiler is limited. For GNU/Linux, SunOS, Cygwin and MinGW we assume that you have at least the following GNU software installed on your computer:

- GNU Autotools:

- `autoconf`,
  - `automake`,
  - `libtool`,
  - `make`, version 3.72.1 or later (check version with `'make --version'`)
- **GCC** - GNU C++ Compiler (g++), version 4.3.x or later (check version with `'gcc --version'`)
  - **Boost** - C++ STL extensions, version 1.35 or later (check version with `'grep "define BOOST_LIB_VERSION" /usr/include/boost/version.hpp'`)
  - **MySQL** - Database client libraries, version 5.0 or later (check version with `'mysql --version'`)
  - **SOCI** - C++ database client library wrapper, version 3.0.0 or later (check version with `'soci-config --version'`)

Optionally, you might need a few additional programs: `Doxygen`, `LaTeX`, `Dvips` and `Ghostscript`, to generate the HTML and PDF documentation.

We strongly recommend that you use recent stable releases of the GCC, if possible. We do not actively work on supporting older versions of the GCC, and they may therefore (without prior notice) become unsupported in future releases of SimCRS.

## 9.4 Basic Installation

Briefly, the shell commands `./cmake .. && make install` should configure, build, and install this package. The following more-detailed instructions are generic; see the `'README'` file for instructions specific to this package. Some packages provide this `'INSTALL'` file but do not implement all of the features documented below. The lack of an optional feature in a given package is not necessarily a bug. More recommendations for GNU packages can be found in the info page corresponding to "Makefile Conventions: (standards)Makefile Conventions".

The `'cmake'` shell script attempts to guess correct values for various system-dependent variables used during compilation. It uses those values to create a `'Makefile'` in each directory of the package. It may also create one or more `'.h'` files containing system-dependent definitions. Finally, it creates a `'CMakeCache.txt'` cache file that you can refer to in the future to recreate the current configuration, and a file `'CMakeFiles'` containing compiler output (useful mainly for debugging `'cmake'`).

It can also use an optional file (typically called `'config.cache'` and enabled with `'-cache-file=config.cache'` or simply `'-C'`) that saves the results of its tests to speed up reconfiguring. Caching is disabled by default to prevent problems with accidental use of stale cache files.

If you need to do unusual things to compile the package, please try to figure out how `'configure'` could check whether to do them, and mail diffs or instructions to the address given in the `'README'` so they can be considered for the next release. If you are using the cache, and at some point `'config.cache'` contains results you don't want to keep, you may remove or edit it.

The file `'CMakeLists.txt'` is used to create the `'CMakefile'`

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.

## 9.5 Compilers and Options

Some systems require unusual options for compilation or linking that the `'cmake'` script does not know about. Run `./cmake -help` for details on some of the pertinent environment variables.

You can give `'cmake'` initial values for configuration parameters by setting variables in the command line or in the environment. Here is an example:

```
./cmake CC=c99 CFLAGS=-g LIBS=-lposix
```

### See Also

[Defining Variables](#) for more details.

## 9.6 Compiling For Multiple Architectures

You can compile the package for more than one kind of computer at the same time, by placing the object files for each architecture in their own directory. To do this, you can use GNU `'make'`. `'cd'` to the directory where you want the object files and executables to go and run the `'configure'` script. `'configure'` automatically checks for the source code in the directory that `'configure'` is in and in `'..'`. This is known as a "VPATH" build.

With a non-GNU `'make'`, it is safer to compile the package for one architecture at a time in the source code directory. After you have installed the package for one architecture, use `'make distclean'` before reconfiguring for another architecture.

On MacOS X 10.5 and later systems, you can create libraries and executables that work on multiple system types-known as "fat" or "universal" binaries-by specifying multiple `'-arch'` options to the compiler but only a single `'-arch'` option to the preprocessor. Like this:

```
./configure CC="gcc -arch i386 -arch x86_64 -arch ppc -arch ppc64" \  
           CXX="g++ -arch i386 -arch x86_64 -arch ppc -arch ppc64" \  
           CPP="gcc -E" CXXCPP="g++ -E"
```

This is not guaranteed to produce working output in all cases, you may have to build one architecture at a time and combine the results using the `'lipo'` tool if you have problems.

## 9.7 Installation Names

By default, `'make install'` installs the package's commands under `'/usr/local/bin'`, include files under `'/usr/local/include'`, etc. You can specify an installation

prefix other than `/usr/local` by giving `configure` the option `-prefix=PREFIX`, where `PREFIX` must be an absolute file name.

You can specify separate installation prefixes for architecture-specific files and architecture-independent files. If you pass the option `-exec-prefix=PREFIX` to `configure`, the package uses `PREFIX` as the prefix for installing programs and libraries. Documentation and other data files still use the regular prefix.

In addition, if you use an unusual directory layout you can give options like `-bindir=DIR` to specify different values for particular kinds of files. Run `configure -help` for a list of the directories you can set and what kinds of files go in them. In general, the default for these options is expressed in terms of `${prefix}`, so that specifying just `-prefix` will affect all of the other directory specifications that were not explicitly provided.

The most portable way to affect installation locations is to pass the correct locations to `configure`; however, many packages provide one or both of the following shortcuts of passing variable assignments to the `make install` command line to change installation locations without having to reconfigure or recompile.

The first method involves providing an override variable for each affected directory. For example, `make install prefix=/alternate/directory` will choose an alternate location for all directory configuration variables that were expressed in terms of `${prefix}`. Any directories that were specified during `configure`, but not in terms of `${prefix}`, must each be overridden at install time for the entire installation to be relocated. The approach of makefile variable overrides for each directory variable is required by the GNU Coding Standards, and ideally causes no recompilation. However, some platforms have known limitations with the semantics of shared libraries that end up requiring recompilation when using this method, particularly noticeable in packages that use GNU Libtool.

The second method involves providing the `DESTDIR` variable. For example, `make install DESTDIR=/alternate/directory` will prepend `/alternate/directory` before all installation names. The approach of `DESTDIR` overrides is not required by the GNU Coding Standards, and does not work on platforms that have drive letters. On the other hand, it does better at avoiding recompilation issues, and works well even when some directory options were not specified in terms of `${prefix}` at `configure` time.

## 9.8 Optional Features

If the package supports it, you can cause programs to be installed with an extra prefix or suffix on their names by giving `cmake` the option `-program-prefix=PREFIX` or `-program-suffix=SUFFIX`.

Some packages pay attention to `-enable-FEATURE` options to `configure`, where `FEATURE` indicates an optional part of the package. They may also pay attention to `-with-PACKAGE` options, where `PACKAGE` is something like `gnu-as` or `x` (for the X Window System). The `README` should mention any `-enable-` and `-with-` options that the package recognizes.

For packages that use the X Window System, `configure` can usually find the X include and library files automatically, but if it doesn't, you can use the `configure` options `-x-includes=DIR` and `-x-libraries=DIR` to specify their locations.

Some packages offer the ability to configure how verbose the execution of `make` will be. For these packages, running `./configure -enable-silent-rules`

sets the default to minimal output, which can be overridden with `'make V=1'`; while running `'./configure --disable-silent-rules'` sets the default to verbose, which can be overridden with `'make V=0'`.

## 9.9 Particular systems

On HP-UX, the default C compiler is not ANSI C compatible. If GNU CC is not installed, it is recommended to use the following options in order to use an ANSI C compiler:

```
./configure CC="cc -Ae -D_XOPEN_SOURCE=500"
```

and if that doesn't work, install pre-built binaries of GCC for HP-UX.

On OSF/1 a.k.a. Tru64, some versions of the default C compiler cannot parse its `<wchar.h>` header file. The option `'-nodtk'` can be used as a workaround. If GNU CC is not installed, it is therefore recommended to try

```
./configure CC="cc"
```

and if that doesn't work, try

```
./configure CC="cc -nodtk"
```

On Solaris, don't put `'/usr/ucb'` early in your `'PATH'`. This directory contains several dysfunctional programs; working variants of these programs are available in `'/usr/bin'`. So, if you need `'/usr/ucb'` in your `'PATH'`, put it *after* `'/usr/bin'`.

On Haiku, software installed for all users goes in `'/boot/common'`, not `'/usr/local'`. It is recommended to use the following options:

```
./cmake -DCMAKE_INSTALL_PREFIX=/boot/common
```

## 9.10 Specifying the System Type

There may be some features `'configure'` cannot figure out automatically, but needs to determine by the type of machine the package will run on. Usually, assuming the package is built to be run on the *same* architectures, `'configure'` can figure that out, but if it prints a message saying it cannot guess the machine type, give it the `'--build=TYPE'` option. TYPE can either be a short name for the system type, such as `'sun4'`, or a canonical name which has the form CPU-COMPANY-SYSTEM

where SYSTEM can have one of these forms:

- OS
- KERNEL-OS

See the file `'config.sub'` for the possible values of each field. If `'config.sub'` isn't included in this package, then this package doesn't need to know the machine type.

If you are *building* compiler tools for cross-compiling, you should use the option `'--target=TYPE'` to select the type of system they will produce code for.

If you want to use a cross compiler, that generates code for a platform different from the build platform, you should specify the "host" platform (i.e., that on which the generated programs will eventually be run) with `'--host=TYPE'`.

## 9.11 Sharing Defaults

If you want to set default values for 'configure' scripts to share, you can create a site shell script called 'config.site' that gives default values for variables like 'CC', 'cache\_file', and 'prefix'. 'configure' looks for 'PREFIX/share/config.site' if it exists, then 'PREFIX/etc/config.site' if it exists. Or, you can set the 'CONFIG\_SITE' environment variable to the location of the site script. A warning: not all 'configure' scripts look for a site script.

## 9.12 Defining Variables

Variables not defined in a site shell script can be set in the environment passed to 'configure'. However, some packages may run configure again during the build, and the customized values of these variables may be lost. In order to avoid this problem, you should set them in the 'configure' command line, using 'VAR=value'. For example:

```
./configure CC=/usr/local2/bin/gcc
```

causes the specified 'gcc' to be used as the C compiler (unless it is overridden in the site shell script).

Unfortunately, this technique does not work for 'CONFIG\_SHELL' due to an Autoconf bug. Until the bug is fixed you can use this workaround:

```
CONFIG_SHELL=/bin/bash /bin/bash ./configure CONFIG_SHELL=/bin/bash
```

## 9.13 'cmake' Invocation

'cmake' recognizes the following options to control how it operates.

- '-help', '-h' print a summary of all of the options to 'cmake', and exit.
- '-help=short', '-help=recursive' print a summary of the options unique to this package's 'configure', and exit. The 'short' variant lists options used only in the top level, while the 'recursive' variant lists options also present in any nested packages.
- '-version', '-V' print the version of Autoconf used to generate the 'configure' script, and exit.
- '-cache-file=FILE' enable the cache: use and save the results of the tests in FILE, traditionally 'config.cache'. FILE defaults to '/dev/null' to disable caching.
- '-config-cache', '-C' alias for '-cache-file=config.cache'.
- '-quiet', '-silent', '-q' do not print messages saying which checks are being made. To suppress all normal output, redirect it to '/dev/null' (any error messages will still be shown).
- '-srcdir=DIR' look for the package's source code in directory DIR. Usually 'configure' can determine that directory automatically.
- '-prefix=DIR' use DIR as the installation prefix.

**See Also**

[Installation Names](#) for more details, including other options available for fine-tuning the installation locations.

- '-no-create', '-n' run the configure checks, but stop before creating any output files.

'cmake' also accepts some other, not widely useful, options. Run 'cmake' -help' for more details.

The 'cmake' script produces an output like this:

```
export LIBSUFFIX_4_CMAKE="-DLIB_SUFFIX=64"
export INSTALL_BASEDIR=/home/user/dev/deliveries
cmake -DCMAKE_INSTALL_PREFIX=${INSTALL_BASEDIR}/simcrs-0.1.0 \
  -DWITH_STDAIR_PREFIX=${INSTALL_BASEDIR}/stdair-stable \
  -DWITH_TRADEMGEM_PREFIX=${INSTALL_BASEDIR}/trademgen-stable \
  -DWITH_TRAVELCCM_PREFIX=${INSTALL_BASEDIR}/travelccm-stable \
  -DWITH_AIRSCHED_PREFIX=${INSTALL_BASEDIR}/airsched-stable \
  -DWITH_AIRRAC_PREFIX=${INSTALL_BASEDIR}/airrac-stable \
  -DWITH_RMOL_PREFIX=${INSTALL_BASEDIR}/rmol-stable \
  -DWITH_AIRINV_PREFIX=${INSTALL_BASEDIR}/airinv-stable \
  -DWITH_SIMFQT_PREFIX=${INSTALL_BASEDIR}/simfqt-stable \
  -DCMAKE_BUILD_TYPE=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: 7a1519ef5b14232c47fe1b1d46db4ae9e65e696d trunk
-- Requires Boost-1.41
-- Boost version: 1.47.0
-- Found the following Boost libraries:
--   regex
--   program_options
--   date_time
--   iostreams
--   serialization
--   filesystem
--   unit_test_framework
--   python
-- Found Boost version: 1.47.0
-- Found BoostWrapper: /usr/include (found suitable version "1.47.0", required is "1.41")
-- Requires Readline without specifying any version
-- Found Readline: /usr/include (found version "6.2")
-- 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 version "5.5.18")
-- Found MySQL version: 5.5.18
-- Requires SOCI-3.0
-- SOCI headers are not buried
-- Found SOCI: /usr/lib64/libsoci_core.so (found suitable version "3.1.0", required is "3.0")
-- Found SOCIMySQL: /usr/lib64/libsoci_mysql.so (found suitable version "3.1.0", required is "3.0")
-- Found SOCI with MySQL back-end support version: 3.1.0
-- Requires StdAir-0.43
-- Found StdAir version: 0.44.3
-- Requires AirSched-0.1
-- Found AirSched version: 0.1.3
-- Requires AirRAC-0.2
-- Found AirRAC version: 0.2.2
-- Requires RMOL-0.25
-- Found RMOL version: 0.25.2
-- Requires AirInv-0.1
-- Found AirInv version: 0.1.2
```

```

-- Requires SimFQT-0.1
-- Found SimFQT version: 0.1.2
-- Requires Doxygen without specifying any version
-- Found Doxygen: /usr/bin/doxygen
-- Found DoxygenWrapper: /usr/bin/doxygen (found version "1.7.5")
-- Found Doxygen version: 1.7.5
-- Had to set the linker language for 'simcrslib' to CXX
-- Test 'CRSTestSuite' to be built with 'CRSTestSuite.cpp'
--
-- =====
-- --- Project Information ---
-- ---
-- PROJECT_NAME ..... : simcrs
-- PACKAGE_PRETTY_NAME ..... : SimCRS
-- PACKAGE ..... : simcrs
-- PACKAGE_NAME ..... : SIMCRS
-- PACKAGE_BRIEF ..... : C++ Simulated Travel-Oriented Distribution System Library
-- PACKAGE_VERSION ..... : 0.5.0
-- GENERIC_LIB_VERSION ..... : 0.5.0
-- GENERIC_LIB_SOVERSION ..... : 0.5
--
-- --- Build Configuration ---
-- ---
-- Modules to build ..... : simcrs
-- Libraries to build/install ..... : simcrslib
-- Binaries to build/install ..... : simcrs
-- Modules to test ..... : simcrs
-- Binaries to test ..... : CRSTestSuitetst
--
-- * Module ..... : simcrs
--   + Layers to build ..... : .;basic;bom;factory;command;service
--   + Dependencies on other layers .. :
--   + Libraries to build/install .... : simcrslib
--   + Executables to build/install .. : simcrs
--   + Tests to perform ..... : CRSTestSuitetst
--
-- BUILD_SHARED_LIBS ..... : ON
-- CMAKE_BUILD_TYPE ..... : Debug
-- * CMAKE_C_FLAGS ..... :
-- * CMAKE_CXX_FLAGS ..... : -Wall -Werror -DBOOST_VERSION=104700
-- * BUILD_FLAGS ..... :
-- * COMPILE_FLAGS ..... :
-- CMAKE_MODULE_PATH ..... : /home/user/dev/sim/simcrs/simcrsgithub/config/
-- CMAKE_INSTALL_PREFIX ..... : /home/user/dev/deliveries/simcrs-0.5.0
--
-- * Doxygen:
--   - DOXYGEN_VERSION ..... : 1.7.5
--   - DOXYGEN_EXECUTABLE ..... : /usr/bin/doxygen
--   - DOXYGEN_DOT_EXECUTABLE ..... : /usr/bin/dot
--   - DOXYGEN_DOT_PATH ..... : /usr/bin
--
-- --- Installation Configuration ---
-- ---
-- INSTALL_LIB_DIR ..... : /home/user/dev/deliveries/simcrs-0.5.0/lib64
-- INSTALL_BIN_DIR ..... : /home/user/dev/deliveries/simcrs-0.5.0/bin
-- CMAKE_INSTALL_RPATH ..... : /home/user/dev/deliveries/simcrs-0.5.0/lib64
-- CMAKE_INSTALL_RPATH_USE_LINK_PATH .. : ON
-- INSTALL_INCLUDE_DIR ..... : /home/user/dev/deliveries/simcrs-0.5.0/include
-- INSTALL_DATA_DIR ..... : /home/user/dev/deliveries/simcrs-0.5.0/share
-- INSTALL_SAMPLE_DIR ..... : /home/user/dev/deliveries/simcrs-0.5.0/share/simcrs/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/user/dev/sim/simcrs/simcrsgithub/README
-- CPACK_RESOURCE_FILE_LICENSE ..... : /home/user/dev/sim/simcrs/simcrsgithub/COPYING

```

```

-- CPACK_GENERATOR ..... : TBZ2
-- CPACK_DEBIAN_PACKAGE_DEPENDS ..... :
-- CPACK_SOURCE_GENERATOR ..... : TBZ2;TGZ
-- CPACK_SOURCE_PACKAGE_FILE_NAME .... : simcrs-0.5.0
--
-- -----
-- ---      External libraries      ---
-- -----
--
-- * Boost:
--   - Boost_VERSION ..... : 104700
--   - Boost_LIB_VERSION ..... : 1_47
--   - Boost_HUMAN_VERSION ..... : 1.47.0
--   - Boost_INCLUDE_DIRS ..... : /usr/include
--   - Boost required components ..... : regex;program_options;date_time;iostreams;serialization;filesystem;un
--   - Boost required libraries ..... : /usr/lib64/libboost_regex-mt.so;/usr/lib64/libboost_iostreams-mt.so;/
--
-- * Readline:
--   - READLINE_VERSION ..... : 6.2
--   - READLINE_INCLUDE_DIR ..... : /usr/include
--   - READLINE_LIBRARY ..... : /usr/lib64/libreadline.so
--
-- * MySQL:
--   - MYSQL_VERSION ..... : 5.5.18
--   - MYSQL_INCLUDE_DIR ..... : /usr/include/mysql
--   - MYSQL_LIBRARIES ..... : /usr/lib64/mysql/libmysqlclient.so
--
-- * SOCI:
--   - SOCI_VERSION ..... : 300100
--   - SOCI_LIB_VERSION ..... : 3_1_0
--   - SOCI_HUMAN_VERSION ..... : 3.1.0
--   - SOCI_INCLUDE_DIR ..... : /usr/include/soci
--   - SOCI_MYSQL_INCLUDE_DIR ..... : /usr/include/soci/mysql
--   - SOCI_LIBRARIES ..... : /usr/lib64/libsoci_core.so
--   - SOCI_MYSQL_LIBRARIES ..... : /usr/lib64/libsoci_mysql.so
--
-- * StdAir:
--   - STDAIR_VERSION ..... : 0.44.3
--   - STDAIR_BINARY_DIRS ..... : /home/user/dev/deliveries/stdair-0.44.3/bin
--   - STDAIR_EXECUTABLES ..... : stdair
--   - STDAIR_LIBRARY_DIRS ..... : /home/user/dev/deliveries/stdair-0.44.3/lib64
--   - STDAIR_LIBRARIES ..... : stdairlib;stdairuiclib
--   - STDAIR_INCLUDE_DIRS ..... : /home/user/dev/deliveries/stdair-0.44.3/include
--   - STDAIR_SAMPLE_DIR ..... : /home/user/dev/deliveries/stdair-0.44.3/share/stdair/samples
--
-- * AirSched:
--   - AIRSCHED_VERSION ..... : 0.1.3
--   - AIRSCHED_BINARY_DIRS ..... : /home/user/dev/deliveries/airsched-0.1.3/bin
--   - AIRSCHED_EXECUTABLES ..... : airsched
--   - AIRSCHED_LIBRARY_DIRS ..... : /home/user/dev/deliveries/airsched-0.1.3/lib64
--   - AIRSCHED_LIBRARIES ..... : airschedlib
--   - AIRSCHED_INCLUDE_DIRS ..... : /home/user/dev/deliveries/airsched-0.1.3/include
--
-- * AirRAC:
--   - AIRRAC_VERSION ..... : 0.2.2
--   - AIRRAC_BINARY_DIRS ..... : /home/user/dev/deliveries/airrac-0.2.2/bin
--   - AIRRAC_EXECUTABLES ..... : airrac
--   - AIRRAC_LIBRARY_DIRS ..... : /home/user/dev/deliveries/airrac-0.2.2/lib64
--   - AIRRAC_LIBRARIES ..... : airraclib
--   - AIRRAC_INCLUDE_DIRS ..... : /home/user/dev/deliveries/airrac-0.2.2/include
--
-- * RMOL:
--   - RMOL_VERSION ..... : 0.25.2
--   - RMOL_BINARY_DIRS ..... : /home/user/dev/deliveries/rmol-0.25.2/bin
--   - RMOL_EXECUTABLES ..... : rmol
--   - RMOL_LIBRARY_DIRS ..... : /home/user/dev/deliveries/rmol-0.25.2/lib
--   - RMOL_LIBRARIES ..... : rmolib
--   - RMOL_INCLUDE_DIRS ..... : /home/user/dev/deliveries/rmol-0.25.2/include
--
-- * AirInv:
--   - AIRINV_VERSION ..... : 0.1.2
--   - AIRINV_BINARY_DIRS ..... : /home/user/dev/deliveries/airinv-0.1.2/bin
--   - AIRINV_EXECUTABLES ..... : airinv;airinv_parseInventory

```

```
-- - AIRINV_LIBRARY_DIRS ..... : /home/user/dev/deliveries/airinv-0.1.2/lib
-- - AIRINV_LIBRARIES ..... : airinvlib
-- - AIRINV_INCLUDE_DIRS ..... : /home/user/dev/deliveries/airinv-0.1.2/include
--
-- * SimFQT:
-- - SIMFQT_VERSION ..... : 0.1.2
-- - SIMFQT_BINARY_DIRS ..... : /home/user/dev/deliveries/simfqt-0.1.2/bin
-- - SIMFQT_EXECUTABLES ..... : simfqt;simfqt_parseFareRules
-- - SIMFQT_LIBRARY_DIRS ..... : /home/user/dev/deliveries/simfqt-0.1.2/lib64
-- - SIMFQT_LIBRARIES ..... : simfqtlib
-- - SIMFQT_INCLUDE_DIRS ..... : /home/user/dev/deliveries/simfqt-0.1.2/include
--
-- Change a value with: cmake -D<Variable>=<Value>
-- =====
--
-- Configuring done
-- Generating done
-- Build files have been written to: /home/user/dev/sim/simcrs/simcrsgithub/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_simcrs
[ 90%] Built target simcrslib
[100%] Built target CRSTestSuitetst
Scanning dependencies of target check_simcrstst
Test project /home/user/dev/sim/simcrs/simcrsgithub/build/test/simcrs
  Start 1: CRSTestSuitetst
1/1 Test #1: CRSTestSuitetst ..... Passed    0.15 sec

100% tests passed, 0 tests failed out of 1

Total Test time (real) = 0.33 sec
[100%] Built target check_simcrstst
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/simcrsgit
rm -rf build && mkdir build
cd build
```

to remove everything.

## 10 Linking with SimCRS

### 10.1 Table of Contents

- [Introduction](#)



- [Dependencies](#)
- [Using the pkg-config command](#)
- [Using the simcrs-config script](#)
- [M4 macro for the GNU Autotools](#)
- [Using SimCRS with dynamic linking](#)

## 10.2 Introduction

There are two convenient methods of linking your programs with the SimCRS library. The first one employs the `'pkg-config'` command (see <http://pkgconfig.freedesktop.org/>), whereas the second one uses `'simcrs-config'` script. These methods are shortly described below.

## 10.3 Dependencies

The SimCRS library depends on several other C++ components.

### 10.3.1 StdAir

Among them, as for now, only StdAir has been packaged. The support for StdAir is taken in charge by a dedicated M4 macro file (namely, `'stdair.m4'`), from the configuration script (generated thanks to `'configure.ac'`).

### 10.3.2 Other Simulation-Related Components

SimCRS, as shown on the diagram below, depends on

- [AirSched](#)
- [SimFQT](#)
- [AirRAC](#)
- [RMOL](#)
- [AirInv](#)
- [AvlCal](#)
- [SimLFS](#)

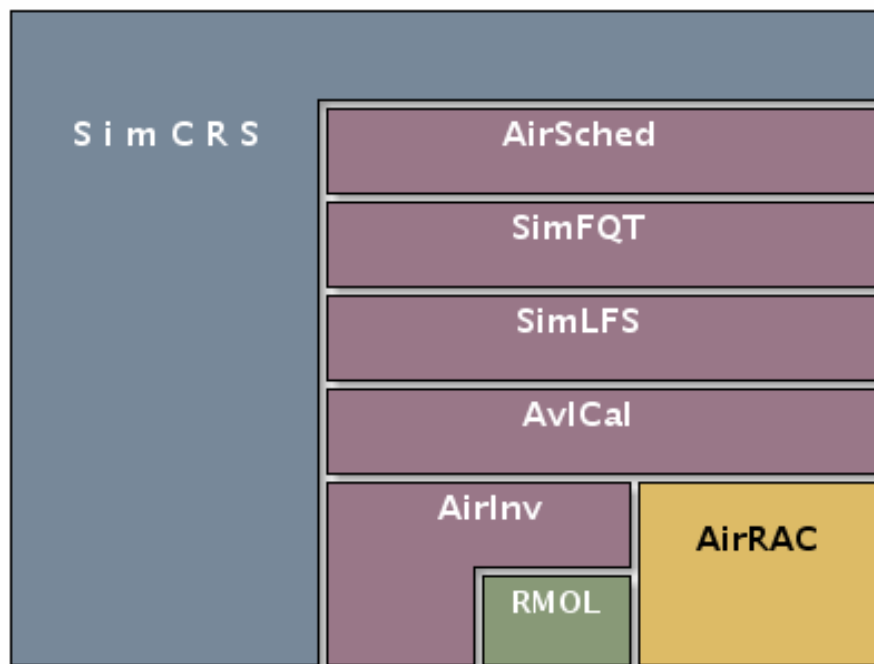


Figure 1: SimCRS Dependencies

## 10.4 Using the pkg-config command

'pkg-config' is a helper tool used when compiling applications and libraries. It helps you insert the correct compiler and linker options. The syntax of the 'pkg-config' is as follows:

```
pkg-config <options> <library_name>
```

For instance, assuming that you need to compile an SimCRS based program 'my\_prog.cpp', you should use the following command:

```
g++ `pkg-config --cflags simcrs` -o my_prog my_prog.cpp `pkg-config --libs  
simcrs`
```

For more information see the 'pkg-config' man pages.

## 10.5 Using the simcrs-config script

SimCRS provides a shell script called `simcrs-config`, which is installed by default in '\$prefix/bin' ('/usr/local/bin') directory. It can be used to simplify compilation and linking of SimCRS 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++ `simcrs-config --cflags` -o my_prog_opt my_prog.cpp `simcrs-config --libs`
```

A list of 'simcrs-config' options can be obtained by typing:

```
simcrs-config --help
```

If the `'simcrs-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
```

## 10.6 M4 macro for the GNU Autotools

A M4 macro file is delivered with SimCRS, namely `'simcrs.m4'`, which can be found in, e.g., `'/usr/share/aclocal'`. When used by a `'configure'` script, thanks to the `'AM_PATH_SimCRS'` macro (specified in the M4 macro file), the following Makefile variables are then defined:

- `'SimCRS_VERSION'` (e.g., defined to `0.23.0`)
- `'SimCRS_CFLAGS'` (e.g., defined to `'-I${prefix}/include'`)
- `'SimCRS_LIBS'` (e.g., defined to `'-L${prefix}/lib -lsimcrs'`)

## 10.7 Using SimCRS with dynamic linking

When using static linking some of the library routines in SimCRS 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 SimCRS library file during your program execution. If you install the SimCRS 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=<SimCRS installation prefix>/lib:$LD_LIBRARY_PATH
```

# 11 Test Rules

This section describes how the functionality of the SimCRS library should be verified. In the `'test/simcrs'` subdirectory, test source files are provided. All functionality should be tested using these test source files.

## 11.1 The Test Source Files

Each new SimCRS module/class should be accompanied with a test source file. The test source file is an implementation in C++ that tests the functionality of a function/class or a group of functions/classes called test suites. The test source file should test relevant parameter settings and input/output relations to guarantee correct functionality of the corresponding classes/functions. The test source files should be maintained using version control and updated whenever new functionality is added to the SimCRS library.

The test source file should print relevant data to a standard output that can be used to verify the functionality. All relevant parameter settings should be tested.

The test source file should be placed in the `'test/simcrs'` subdirectory and should have a name ending with `'TestSuite.cpp'`.

## 11.2 The Reference File

Consider a test source file named `'YieldTestSuite.cpp'`. A reference file named `'YieldTestSuite.ref'` should accompany the test source file. The reference file contains a reference printout of the standard output generated when running the test program. The reference file should be maintained using version control and updated according to the test source file.

## 11.3 Testing SimCRS Library

One can compile and execute all test programs from the ``test/simcrs'` sub-directory by typing:

```
% make check
```

after successful compilation of the SimCRS library.

## 12 Users Guide

### 12.1 Table of Contents

- [Introduction](#)
- [Get Started](#)
  - [Get the SimCRS library](#)
  - [Build the SimCRS project](#)
  - [Build and Run the Tests](#)
  - [Install the SimCRS Project \(Binaries, Documentation\)](#)
- [Input file of SimCRS 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](#)

### 12.2 Introduction

The `SimCRS` library contains classes for airline business management. This document does not cover all the aspects of the `SimCRS` library. It does however explain the most important things you need to know in order to start using `SimCRS`.

### 12.3 Get Started

#### 12.3.1 Get the SimCRS library

Clone locally the full `Git` project:

```
cd ~
mkdir -p dev/sim
cd ~/dev/sim
git clone git://simcrs.git.sourceforge.net/gitroot/simcrs/simcrs simcrsgit
cd simcrsgit
git checkout trunk
```

### 12.3.2 Build the SimCRS project

Link with StdAir, create the distribution package (say, 0.5.0) and compile using the following commands:

```
cd ~/dev/sim/simcrsgit
rm -rf build && mkdir -p build
cd build
cmake -DCMAKE_INSTALL_PREFIX=~/.dev/deliveries/simcrs-0.5.0 \
  -DWITH_STDAIR_PREFIX=~/.dev/deliveries/stdair-stable \
  -DCMAKE_BUILD_TYPE:String=Debug -DINSTALL_DOC:BOOL=ON ..
make
```

### 12.3.3 Build and Run the Tests

After building the SimCRS project, the following commands run the tests:

```
cd ~/dev/sim/simcrsgit
cd build
make check
```

As a result, you should obtain a similar report:

```
[ 0%] Built target hdr_cfg_simcrs
[ 96%] Built target simcrslib
[100%] Built target AirlineScheduleTestSuitetst
Scanning dependencies of target check_simcrstst
Test project /home/dan/dev/sim/simcrs/simcrsgithub/build/test/simcrs
  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_simcrstst
Scanning dependencies of target check
[100%] Built target check
```

### 12.3.4 Install the SimCRS Project (Binaries, Documentation)

After the step [Build the SimCRS project](#), to install the library and its header files, type:

```
cd ~/dev/sim/simcrsgit
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/simcrs-0.5.0
```

To generate the SimCRS project documentation, the commands are:

```
cd ~/dev/sim/simcrsgit
cd build
make doc
```

The SimCRS project documentation is available in the following formats: HTML, LaTeX. Those documents are available in a subdirectory:

```
cd ~/dev/sim/simcrsgit
cd build
cd doc
```

## 12.4 Input file of SimCRS 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 `SIM-CRS::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](#).

## 12.5 The schedule BOM Tree

The schedule-related Business Object Model (BOM) tree is a structure allowing to store all the `SIMCRS::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 SimCRS 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.

### 12.5.1 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 `SIMCRS::SIMCRS_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 `SIMCRS::SIMCRS_Service::buildSampleBom()` method:

```
void buildSampleBom ();
```

or can be constructed using the schedule input file described above thanks to the `SIMCRS::SIMCRS_Service::parseAndLoad (const stdair::Filename_T& iScheduleInputFilename,`

```
void parseAndLoad (const stdair::Filename_T& iScheduleInputFilename,
```

### 12.5.2 Display of the schedule BOM tree

#### Note

That feature (of BOM tree display) has not been implemented yet. Do not hesitate to [open a ticket](#) if you would like to have it implemented more quickly.

The schedule BOM tree can be displayed as done in the `batches::simcrs.cpp` program:

When the default BOM tree is used (`-b/-builtin` option of the main program `simcrs.cpp`), the schedule BOM tree display (for now, corresponding to `schedule01.csv` parsed by `SIMCRS::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, GAV, ACP, ETB, BidPrice,
SQ11 2010-Jan-15, SIN-BKK 2010-Jan-15, Y, 300, 300, 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, ETB, ClassAvl, RevAvl, SegAvl,
SQ11 2010-Jan-15, SIN-BKK 2010-Jan-15, Y, 1, Y, 300 (0), 0, 0, 0, 2, 0 (0), 0,
0, 0, 0, 0, 0,
SQ11 2010-Jan-15, SIN-BKK 2010-Jan-15, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Jan-16
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
Elapsed, Distance, Capacity,
SQ11 2010-Jan-16, SIN-BKK, 2010-Jan-16, 08:20:00, 2010-Jan-16, 11:00:00, 07:40:
00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhycAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ11 2010-Jan-16, SIN-BKK 2010-Jan-16, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 300
, 9, 1.83244e-319, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, 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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ11 2010-Jan-18, SIN-BKK 2010-Jan-18, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300
, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-18, SIN-BKK 2010-Jan-18, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-18, SIN-BKK 2010-Jan-18, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ11 2010-Jan-19, SIN-BKK 2010-Jan-19, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300
, 9, 0, 0, 0, 0, 0,
*****
*****
```

```
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-19, SIN-BKK 2010-Jan-19, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-19, SIN-BKK 2010-Jan-19, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
  GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
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,
SQ11 2010-Jan-20, SIN-BKK, 2010-Jan-20, 08:20:00, 2010-Jan-20, 11:00:00, 07:40:
  00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
  CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ11 2010-Jan-20, SIN-BKK 2010-Jan-20, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300
  , 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-20, SIN-BKK 2010-Jan-20, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-20, SIN-BKK 2010-Jan-20, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
  GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQ11 2010-Jan-20, SIN-BKK 2010-Jan-20, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0,
  0, 0, 0, 0, 0,
SQ11 2010-Jan-20, SIN-BKK 2010-Jan-20, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0,
  0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Jan-21
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
  Elapsed, Distance, Capacity,
SQ11 2010-Jan-21, SIN-BKK, 2010-Jan-21, 08:20:00, 2010-Jan-21, 11:00:00, 07:40:
  00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
  CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ11 2010-Jan-21, SIN-BKK 2010-Jan-21, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300
  , 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
```

```

SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-21, SIN-BKK 2010-Jan-21, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-21, SIN-BKK 2010-Jan-21, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
      GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQ11 2010-Jan-21, SIN-BKK 2010-Jan-21, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
SQ11 2010-Jan-21, SIN-BKK 2010-Jan-21, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Jan-22
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
      Elapsed, Distance, Capacity,
SQ11 2010-Jan-22, SIN-BKK, 2010-Jan-22, 08:20:00, 2010-Jan-22, 11:00:00, 07:40:
      00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhycAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
      CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ11 2010-Jan-22, SIN-BKK 2010-Jan-22, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ11 2010-Jan-23, SIN-BKK 2010-Jan-23, Y, 300, 300, 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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ11 2010-Jan-25, SIN-BKK 2010-Jan-25, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
      GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
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,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
      GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ11 2010-Jan-28, SIN-BKK 2010-Jan-28, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300
, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-28, SIN-BKK 2010-Jan-28, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-28, SIN-BKK 2010-Jan-28, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, ETB, ClassAvl, RevAvl, SegAvl,
SQ11 2010-Jan-29, SIN-BKK 2010-Jan-29, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
SQ11 2010-Jan-29, SIN-BKK 2010-Jan-29, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Jan-30
*****
*****
```

```
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
Elapsed, Distance, Capacity,
SQ11 2010-Jan-30, SIN-BKK, 2010-Jan-30, 08:20:00, 2010-Jan-30, 11:00:00, 07:40:
00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ11 2010-Jan-30, SIN-BKK 2010-Jan-30, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300
, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-30, SIN-BKK 2010-Jan-30, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-30, SIN-BKK 2010-Jan-30, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQ11 2010-Jan-30, SIN-BKK 2010-Jan-30, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
SQ11 2010-Jan-30, SIN-BKK 2010-Jan-30, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Jan-31
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
Elapsed, Distance, Capacity,
SQ11 2010-Jan-31, SIN-BKK, 2010-Jan-31, 08:20:00, 2010-Jan-31, 11:00:00, 07:40:
00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ11 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,
SQ11 2010-Jan-31, SIN-BKK 2010-Jan-31, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-31, SIN-BKK 2010-Jan-31, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQ11 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,
SQ11 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, GAV, ACP, ETB, BidPrice,
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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ11 2010-Feb-05, SIN-BKK 2010-Feb-05, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300
, 9, 0, 0, 0, 0, 0,
*****

```

```
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL1 2010-Feb-05, SIN-BKK 2010-Feb-05, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 2010-Feb-05, SIN-BKK 2010-Feb-05, Y, 2, 0, 0, 0, 0, 300, 0,
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
      GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQL1 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,
SQL1 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: SQL1, 2010-Feb-06
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
      Elapsed, Distance, Capacity,
SQL1 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, GAV, ACP, ETB, BidPrice,
SQL1 2010-Feb-06, SIN-BKK 2010-Feb-06, 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-06, SIN-BKK 2010-Feb-06, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 2010-Feb-06, SIN-BKK 2010-Feb-06, Y, 2, 0, 0, 0, 0, 300, 0,
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
      GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQL1 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, GAV, ACP, ETB, BidPrice,
SQL1 2010-Feb-07, SIN-BKK 2010-Feb-07, 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-07, SIN-BKK 2010-Feb-07, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-07, SIN-BKK 2010-Feb-07, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
      GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQ11 2010-Feb-07, SIN-BKK 2010-Feb-07, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
SQ11 2010-Feb-07, SIN-BKK 2010-Feb-07, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-08
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
      Elapsed, Distance, Capacity,
SQ11 2010-Feb-08, SIN-BKK, 2010-Feb-08, 08:20:00, 2010-Feb-08, 11:00:00, 07:40:
      00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
      CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ11 2010-Feb-08, SIN-BKK 2010-Feb-08, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300
      , 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-08, SIN-BKK 2010-Feb-08, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-08, SIN-BKK 2010-Feb-08, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
      GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQ11 2010-Feb-08, SIN-BKK 2010-Feb-08, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
SQ11 2010-Feb-08, SIN-BKK 2010-Feb-08, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-09
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
      Elapsed, Distance, Capacity,
SQ11 2010-Feb-09, SIN-BKK, 2010-Feb-09, 08:20:00, 2010-Feb-09, 11:00:00, 07:40:
      00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
      CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ11 2010-Feb-09, SIN-BKK 2010-Feb-09, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300
      , 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-09, SIN-BKK 2010-Feb-09, Y, 1, 0, 0, 0, 0, 300, 0,
```

```
SQL1 2010-Feb-09, SIN-BKK 2010-Feb-09, Y, 2, 0, 0, 0, 0, 300, 0,
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
  GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, 300, 0,
SQL1 2010-Feb-10, SIN-BKK 2010-Feb-10, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
  GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQL1 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,
SQL1 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: SQL1, 2010-Feb-11
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
  Elapsed, Distance, Capacity,
SQL1 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, GAV, ACP, ETB, BidPrice,
SQL1 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,
SQL1 2010-Feb-11, SIN-BKK 2010-Feb-11, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ11 2010-Feb-12, SIN-BKK 2010-Feb-12, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ11 2010-Feb-13, SIN-BKK 2010-Feb-13, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ11 2010-Feb-14, SIN-BKK 2010-Feb-14, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300
, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-14, SIN-BKK 2010-Feb-14, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-14, SIN-BKK 2010-Feb-14, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ11 2010-Feb-15, SIN-BKK 2010-Feb-15, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300
, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-15, SIN-BKK 2010-Feb-15, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-15, SIN-BKK 2010-Feb-15, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ11 2010-Feb-16, SIN-BKK 2010-Feb-16, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300
, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-16, SIN-BKK 2010-Feb-16, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-16, SIN-BKK 2010-Feb-16, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, ETB, ClassAvl, RevAvl, SegAvl,
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,
```

```
SQL11 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, GAV, ACP, ETB, BidPrice,
SQL11 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,
SQL11 2010-Feb-18, SIN-BKK 2010-Feb-18, Y, 1, 0, 0, 0, 0, 300, 0,
SQL11 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, ETB, ClassAvl, RevAvl, SegAvl,
SQL11 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,
SQL11 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: SQL11, 2010-Feb-19
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
Elapsed, Distance, Capacity,
SQL11 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, GAV, ACP, ETB, BidPrice,
SQL11 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,
SQL11 2010-Feb-19, SIN-BKK 2010-Feb-19, Y, 1, 0, 0, 0, 0, 300, 0,
SQL11 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, ETB, ClassAvl, RevAvl, SegAvl,
SQL11 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,
SQL11 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: SQL11, 2010-Feb-20
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
Elapsed, Distance, Capacity,
SQL11 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, GAV, ACP, ETB, BidPrice,
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,
SQ11 2010-Feb-20, SIN-BKK 2010-Feb-20, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQ11 2010-Feb-20, SIN-BKK 2010-Feb-20, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
SQ11 2010-Feb-20, SIN-BKK 2010-Feb-20, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-21
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
Elapsed, Distance, Capacity,
SQ11 2010-Feb-21, SIN-BKK, 2010-Feb-21, 08:20:00, 2010-Feb-21, 11:00:00, 07:40:
00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhycAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ11 2010-Feb-21, SIN-BKK 2010-Feb-21, 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-21, SIN-BKK 2010-Feb-21, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-21, SIN-BKK 2010-Feb-21, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQ11 2010-Feb-21, SIN-BKK 2010-Feb-21, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
SQ11 2010-Feb-21, SIN-BKK 2010-Feb-21, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-22
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
Elapsed, Distance, Capacity,
SQ11 2010-Feb-22, SIN-BKK, 2010-Feb-22, 08:20:00, 2010-Feb-22, 11:00:00, 07:40:
00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhycAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ11 2010-Feb-22, SIN-BKK 2010-Feb-22, 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-22, SIN-BKK 2010-Feb-22, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 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, ETB, ClassAvl, RevAvl, SegAvl,
SQ11 2010-Feb-22, SIN-BKK 2010-Feb-22, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
SQ11 2010-Feb-22, SIN-BKK 2010-Feb-22, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-23
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
Elapsed, Distance, Capacity,
SQ11 2010-Feb-23, SIN-BKK, 2010-Feb-23, 08:20:00, 2010-Feb-23, 11:00:00, 07:40:
00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ11 2010-Feb-23, SIN-BKK 2010-Feb-23, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300
, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-23, SIN-BKK 2010-Feb-23, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 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, ETB, ClassAvl, RevAvl, SegAvl,
SQ11 2010-Feb-23, SIN-BKK 2010-Feb-23, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
SQ11 2010-Feb-23, SIN-BKK 2010-Feb-23, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-24
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
Elapsed, Distance, Capacity,
SQ11 2010-Feb-24, SIN-BKK, 2010-Feb-24, 08:20:00, 2010-Feb-24, 11:00:00, 07:40:
00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ11 2010-Feb-24, SIN-BKK 2010-Feb-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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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,
```

```
SQL1 2010-Feb-26, SIN-BKK 2010-Feb-26, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 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, ETB, ClassAvl, RevAvl, SegAvl,
SQL1 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,
SQL1 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: SQL1, 2010-Feb-27
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
      Elapsed, Distance, Capacity,
SQL1 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, GAV, ACP, ETB, BidPrice,
SQL1 2010-Feb-27, SIN-BKK 2010-Feb-27, 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-27, SIN-BKK 2010-Feb-27, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 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, ETB, ClassAvl, RevAvl, SegAvl,
SQL1 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,
SQL1 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: SQL1, 2010-Feb-28
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
      Elapsed, Distance, Capacity,
SQL1 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, GAV, ACP, ETB, BidPrice,
SQL1 2010-Feb-28, SIN-BKK 2010-Feb-28, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 300
      , 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL1 2010-Feb-28, SIN-BKK 2010-Feb-28, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, 1.05119e-153, 2.78986e+179, 0, 200, 9, 3.66962e-62, 1
  .0854e-71, 6.74783e-67, 6.9835e-77, 0,
*****
*****
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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, 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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Jan-17, SIN-HND 2010-Jan-17, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 200
, 9, 2.39291e-319, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Jan-18, SIN-HND 2010-Jan-18, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 200
, 9, 2.14469e-319, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, 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, 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, GAV, ACP, ETB, BidPrice,
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, 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, ETB, ClassAvl, RevAvl, SegAvl,
SQ12 2010-Jan-19, SIN-HND 2010-Jan-19, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
SQ12 2010-Jan-19, SIN-HND 2010-Jan-19, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Jan-20
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
Elapsed, Distance, Capacity,
SQ12 2010-Jan-20, SIN-HND, 2010-Jan-20, 09:20:00, 2010-Jan-20, 12:00:00, 07:40:
00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ12 2010-Jan-20, SIN-HND 2010-Jan-20, Y, 200, 200, 0, 0, 0, 0, 0, 0, 200
, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Jan-20, SIN-HND 2010-Jan-20, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Jan-20, SIN-HND 2010-Jan-20, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Jan-21, SIN-HND 2010-Jan-21, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Jan-22, SIN-HND 2010-Jan-22, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Jan-23, SIN-HND 2010-Jan-23, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Jan-24, SIN-HND 2010-Jan-24, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
```

```
SQL12 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,
SQL12 2010-Jan-25, SIN-HND 2010-Jan-25, Y, 1, 0, 0, 0, 0, 200, 0,
SQL12 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, ETB, ClassAvl, RevAvl, SegAvl,
SQL12 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,
SQL12 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: SQL12, 2010-Jan-26
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
Elapsed, Distance, Capacity,
SQL12 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, GAV, ACP, ETB, BidPrice,
SQL12 2010-Jan-26, SIN-HND 2010-Jan-26, 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,
SQL12 2010-Jan-26, SIN-HND 2010-Jan-26, Y, 1, 0, 0, 0, 0, 200, 0,
SQL12 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, ETB, ClassAvl, RevAvl, SegAvl,
SQL12 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,
SQL12 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: SQL12, 2010-Jan-27
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
Elapsed, Distance, Capacity,
SQL12 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, GAV, ACP, ETB, BidPrice,
SQL12 2010-Jan-27, SIN-HND 2010-Jan-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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 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,
SQ12 2010-Jan-30, SIN-HND 2010-Jan-30, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 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, ETB, ClassAvl, RevAvl, SegAvl,
SQ12 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,
SQ12 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: SQ12, 2010-Jan-31
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
Elapsed, Distance, Capacity,
SQ12 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, GAV, ACP, ETB, BidPrice,
SQ12 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,
SQ12 2010-Jan-31, SIN-HND 2010-Jan-31, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 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, ETB, ClassAvl, RevAvl, SegAvl,
SQ12 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,
SQ12 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: SQ12, 2010-Feb-01
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
    Elapsed, Distance, Capacity,
SQ12 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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-01, SIN-HND 2010-Feb-01, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 200
    , 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-02, SIN-HND 2010-Feb-02, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 200
    , 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-02, SIN-HND 2010-Feb-02, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-02, SIN-HND 2010-Feb-02, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
    GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
```

```

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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-03, SIN-HND 2010-Feb-03, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200
, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-03, SIN-HND 2010-Feb-03, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-03, SIN-HND 2010-Feb-03, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-05, SIN-HND 2010-Feb-05, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-06, SIN-HND 2010-Feb-06, 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, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-07, SIN-HND 2010-Feb-07, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-08, SIN-HND 2010-Feb-08, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-09, SIN-HND 2010-Feb-09, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200
, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-09, SIN-HND 2010-Feb-09, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-09, SIN-HND 2010-Feb-09, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-10, SIN-HND 2010-Feb-10, Y, 200, 200, 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, ETB, ClassAvl, RevAvl, SegAvl,
SQ12 2010-Feb-10, SIN-HND 2010-Feb-10, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
SQ12 2010-Feb-10, SIN-HND 2010-Feb-10, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-11
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
Elapsed, Distance, Capacity,
SQ12 2010-Feb-11, SIN-HND, 2010-Feb-11, 09:20:00, 2010-Feb-11, 12:00:00, 07:40:
00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhycAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
```

```
CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-11, SIN-HND 2010-Feb-11, 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-11, SIN-HND 2010-Feb-11, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-11, SIN-HND 2010-Feb-11, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQ12 2010-Feb-11, SIN-HND 2010-Feb-11, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
SQ12 2010-Feb-11, SIN-HND 2010-Feb-11, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-12
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
Elapsed, Distance, Capacity,
SQ12 2010-Feb-12, SIN-HND, 2010-Feb-12, 09:20:00, 2010-Feb-12, 12:00:00, 07:40:
00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-12, SIN-HND 2010-Feb-12, 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-12, SIN-HND 2010-Feb-12, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-12, SIN-HND 2010-Feb-12, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQ12 2010-Feb-12, SIN-HND 2010-Feb-12, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
SQ12 2010-Feb-12, SIN-HND 2010-Feb-12, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0,
0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-13
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
Elapsed, Distance, Capacity,
SQ12 2010-Feb-13, SIN-HND, 2010-Feb-13, 09:20:00, 2010-Feb-13, 12:00:00, 07:40:
00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-13, SIN-HND 2010-Feb-13, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-16, SIN-HND 2010-Feb-16, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-17, SIN-HND 2010-Feb-17, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-19, SIN-HND 2010-Feb-19, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
      GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-22, SIN-HND 2010-Feb-22, 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-22, SIN-HND 2010-Feb-22, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-22, SIN-HND 2010-Feb-22, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-23, SIN-HND 2010-Feb-23, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-24, SIN-HND 2010-Feb-24, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-25, SIN-HND 2010-Feb-25, Y, 200, 200, 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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
SQ12 2010-Feb-26, SIN-HND 2010-Feb-26, 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-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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, ETB, ClassAvl, RevAvl, SegAvl,
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, GAV, ACP, ETB, BidPrice,
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, ETB, ClassAvl, RevAvl, SegAvl,
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,
*****

```

## 12.6 Exploring the Predefined BOM Tree

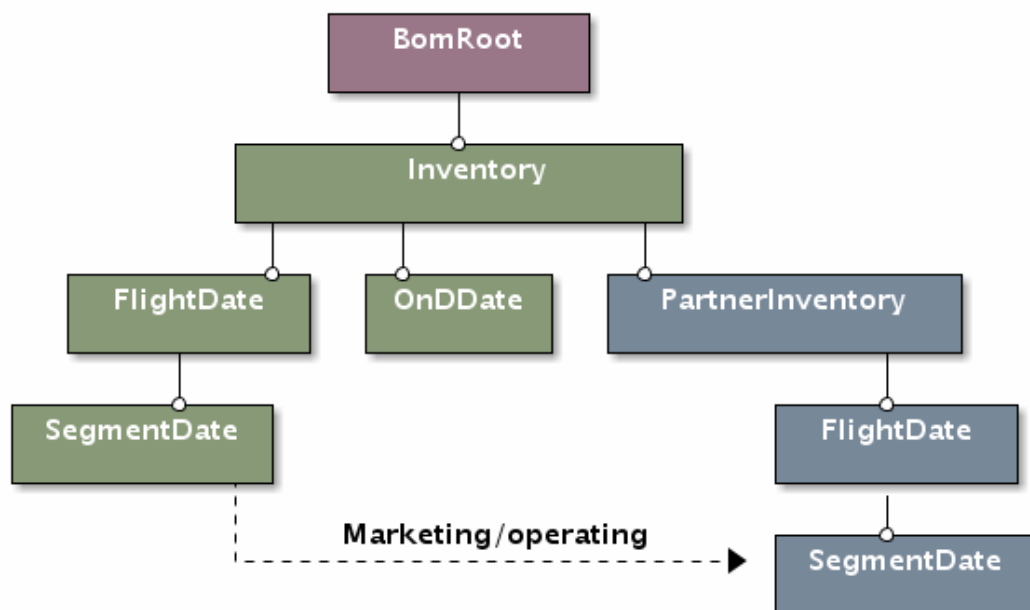


Figure 2: SimCRS BOM tree

SimCRS predefines a BOM (Business Object Model) tree specific to the airline IT arena.

### 12.6.1 Airline Network BOM Tree

- SIMCRS::ReachableUniverse
- SIMCRS::OriginDestinationSet

- `SIMCRS::SegmentPathPeriod`

### 12.6.2 Airline Schedule BOM Tree

- `stdair::Inventory`
- `stdair::FlightPeriod`
- `stdair::SegmentPeriod`
- `stdair::OnDPeriod`

## 12.7 Extending the BOM Tree

### 12.8 The travel solution calculation procedure

The project SimCRS aims at calculating a list of `travel solutions` for every incoming `booking request`.

## 13 Supported Systems

### 13.1 Table of Contents

- [Introduction](#)
- [.1 SimCRS 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 SimCRS 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 SimCRS External](#)
    - \* [Microsoft Windows XP with MS Visual C++ and Intel MKL](#)
  - [Unix Systems](#)
    - \* [SunOS 5.9 with SimCRS External](#)
- [SimCRS 3.9.1](#)
- [SimCRS 3.9.0](#)
- [SimCRS 3.8.1](#)

## 13.2 Introduction

This page is intended to provide a list of SimCRS supported systems, i.e. the systems on which configuration, installation and testing process of the SimCRS 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 SimCRS library on a system not mentioned below, please let us know, so we could update this database.

## 14 SimCRS Supported Systems (Previous Releases)

### 14.1 SimCRS 3.9.1

### 14.2 SimCRS 3.9.0

### 14.3 SimCRS 3.8.1

## 15 Tutorials

### 15.1 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](#)

### 15.2 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.

### 15.3 Your first networkBuilde

#### 15.3.1 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 `AIRSCHEd_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](#)):

### 15.3.2 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.

## 15.4 Network building with an input file

### 15.4.1 How to build a network input file?

The objective here is to build a network input file to network build the default travel solution list built using:

This travel solution list, reduced to a singleton, can be displayed as done before:

We deduce:

- we need a network rule whose origin-destination couple is "LHR, SYD".
- the date range must include the date "2011-06-10".

- the time range must include the time "21:45".
- the airline operating is "BA", so it must be the airline pricing.

We can deduce a part of our network rule file :

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:

#### 15.4.2 Building the BOM tree with an input file

The steps are the same as before [Summary of the different steps](#) except the bom tree must be built using the network input file :

#### 15.4.3 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.

## 16 Command-Line Test to Demonstrate How To Test the SimCRS Project

```
*/
// //////////////////////////////////////
// Import section
// //////////////////////////////////////
// STL
```

```

#include <sstream>
#include <fstream>
#include <string>
#include <cmath>
// Boost Unit Test Framework (UTF)
#define BOOST_TEST_DYN_LINK
#define BOOST_TEST_MAIN
#define BOOST_TEST_MODULE CRSTestSuite
#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>
// SimFQT
#include <simfqt/SIMFQT_Types.hpp>
// SimCRS
#include <simcrs/SIMCRS_Service.hpp>
#include <simcrs/config/simcrs-paths.hpp>

namespace boost_utf = boost::unit_test;

// (Boost) Unit Test XML Report
std::ofstream utfReportStream ("CRSTestSuite_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() {
    }
};

// //////////// 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 (simcrs_simple_simulation_test) {

    // CRS code
    const SIMCRS::CRSCode_T lCRSCode ("1P");

    // Schedule input filename
    const stdair::Filename_T lScheduleInputFilename (STDAIR_SAMPLE_DIR
                                                    "/rds01/schedule.csv");

    // O&D input filename
    const stdair::Filename_T lOnDInputFilename (STDAIR_SAMPLE_DIR
                                                "/ond01.csv");

    // Yield input filename
    const stdair::Filename_T lYieldInputFilename (STDAIR_SAMPLE_DIR
                                                  "/rds01/yield.csv");

    // Fare input filename
    const stdair::Filename_T lFareInputFilename (STDAIR_SAMPLE_DIR
                                                  "/rds01/fare.csv");

    // Check that the file path given as input corresponds to an actual file
    bool doesExistAndIsReadable =
        stdair::BasFileMgr::doesExistAndIsReadable (lScheduleInputFilename);
    BOOST_CHECK_MESSAGE (doesExistAndIsReadable == true,
        "The '" << lScheduleInputFilename
        << "' input file can not be open and read");

    // Check that the file path given as input corresponds to an actual file
    doesExistAndIsReadable =
        stdair::BasFileMgr::doesExistAndIsReadable (lOnDInputFilename);
    BOOST_CHECK_MESSAGE (doesExistAndIsReadable == true,
        "The '" << lOnDInputFilename
        << "' input file can not be open and read");

    // Check that the file path given as input corresponds to an actual file
    doesExistAndIsReadable =

```

```

    stdair::BasFileMgr::doesExistAndIsReadable (lYieldInputFilename);
BOOST_CHECK_MESSAGE (doesExistAndIsReadable == true,
    "The '" << lYieldInputFilename
    << "' input file can not be open and read");

// Check that the file path given as input corresponds to an actual file
doesExistAndIsReadable =
    stdair::BasFileMgr::doesExistAndIsReadable (lFareInputFilename);
BOOST_CHECK_MESSAGE (doesExistAndIsReadable == true,
    "The '" << lFareInputFilename
    << "' input file can not be open and read");

// Output log File
const stdair::Filename_T lLogFilename ("CRSTestSuite.log");

// Set the log parameters
std::ofstream logOutputFile;
// Open and clean the log outputfile
logOutputFile.open (lLogFilename.c_str());
logOutputFile.clear();

// Initialise the list of classes/buckets
const stdair::BasLogParams lLogParams (stdair::LOG::DEBUG, logOutputFile);
SIMCRS::SIMCRS_Service simcrsService (lLogParams,
    lCRSCode);

// Build the BOM tree from parsing input files
const SIMFQT::FareFilePath lFareFilePath (lFareInputFilename);
const AIRRAC::YieldFilePath lYieldFilePath (lYieldInputFilename);
simcrsService.parseAndLoad (lScheduleInputFilename, lOnDInputFilename,
    lYieldFilePath, lFareFilePath);

// Create an empty booking request structure
// TODO: fill the booking request structure from the input parameters
const stdair::AirportCode_T lOrigin ("SIN");
const stdair::AirportCode_T lDestination ("BKK");
const stdair::AirportCode_T lPOS ("SIN");
const stdair::Date_T lPreferredDepartureDate(2011, boost::gregorian::Jan, 31)
;
const stdair::Date_T lRequestDate (2011, boost::gregorian::Jan, 22);
const stdair::Duration_T lRequestTime (boost::posix_time::hours(10));
const stdair::DateTime_T lRequestDateTime (lRequestDate, lRequestTime);
const stdair::CabinCode_T lPreferredCabin ("Eco");
const stdair::PartySize_T lPartySize (3);
const stdair::ChannelLabel_T lChannel ("IN");
const stdair::TripType_T lTripType ("RI");
const stdair::DayDuration_T lStayDuration (7);
const stdair::FrequentFlyer_T lFrequentFlyerType ("M");
const stdair::Duration_T lPreferredDepartureTime (boost::posix_time::hours(10)
));
const stdair::WTP_T lWTP (1000.0);
const stdair::PriceValue_T lValueOfTime (100.0);
const stdair::BookingRequestStruct lBookingRequest (lOrigin, lDestination,
    lPOS,
    lPreferredDepartureDate,
    lRequestDateTime,
    lPreferredCabin,
    lPartySize, lChannel,
    lTripType, lStayDuration,
    lFrequentFlyerType,
    lPreferredDepartureTime,
    lWTP, lValueOfTime);

stdair::TravelSolutionList_T lTravelSolutionList =
    simcrsService.calculateSegmentPathList (lBookingRequest);

// Price the travel solution
simcrsService.fareQuote (lBookingRequest, lTravelSolutionList);

//
const unsigned int lNbOfTravelSolutions = lTravelSolutionList.size();

// \todo change the expected number of travel solutions to the actual number
const unsigned int lExpectedNbOfTravelSolutions = 1;

// DEBUG
std::ostringstream oMessageKeptTS;
oMessageKeptTS << "The number of travel solutions for the booking request '"
    << lBookingRequest.describe() << "' is actually "
    << lNbOfTravelSolutions << ". That number is expected to be "
    << lExpectedNbOfTravelSolutions << ".";
STDAIR_LOG_DEBUG (oMessageKeptTS.str());

BOOST_CHECK_EQUAL (lNbOfTravelSolutions, lExpectedNbOfTravelSolutions);

BOOST_CHECK_MESSAGE (lNbOfTravelSolutions == lExpectedNbOfTravelSolutions,
    oMessageKeptTS.str());

```



```

stdair::TravelSolutionStruct& lTravelSolution = lTravelSolutionList.front();

const stdair::FareOptionList_T& lFareOptionList =
    lTravelSolution.getFareOptionList();

stdair::FareOptionStruct lFareOption = lFareOptionList.front();
lTravelSolution.setChosenFareOption (lFareOption);

const unsigned int lExpectedPrice = 400;

// DEBUG
std::ostream oMessageKeptFare;
oMessageKeptFare
    << "The price given by the fare quoter for the booking request: '"
    << lBookingRequest.describe() << "' and travel solution: '"
    << lTravelSolution.describe() << "' is actually '" << lFareOption.getFare()
    << " Euros. It is expected to be '" << lExpectedPrice << " Euros.";
STDAIR_LOG_DEBUG (oMessageKeptFare.str());

BOOST_CHECK_EQUAL (std::floor (lFareOption.getFare() + 0.5), lExpectedPrice);

BOOST_CHECK_MESSAGE (std::floor (lFareOption.getFare() + 0.5)
    == lExpectedPrice, oMessageKeptFare.str());

// DEBUG
STDAIR_LOG_DEBUG ("A booking will now (attempted to) be made on the '"
    "travel solution '" << lTravelSolution.describe()
    << "', for a party size of '" << lPartySize << "."");

const bool isSellSuccessful =
    simcrsService.sell (lTravelSolution, lPartySize);
//BOOST_CHECK_NO_THROW ();

// DEBUG
std::ostream oMessageSell;
const std::string isSellSuccessfulStr = (isSellSuccessful == true)?"Yes":"No"
;
oMessageSell << "Was the sell successful? Answer: '" << isSellSuccessfulStr;
STDAIR_LOG_DEBUG (oMessageSell.str());

BOOST_CHECK_EQUAL (isSellSuccessful, true);

BOOST_CHECK_MESSAGE (isSellSuccessful == true, oMessageSell.str());

// Close the log file
logOutputFile.close();
}

// End the test suite
BOOST_AUTO_TEST_SUITE_END()

/*!

```

## 17 Namespace Index

### 17.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<b>AIRINV</b>	<b>84</b>
<b>SIMCRS</b>	<b>84</b>
<b>stdair</b>	
<b>Forward declarations</b>	<b>85</b>

## 18 Class Index

### 18.1 Class Hierarchy

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

```
std::basic_fstream< char >
```

std::basic_fstream< wchar_t >	
std::basic_ifstream< char >	
std::basic_ifstream< wchar_t >	
std::basic_ios< char >	
std::basic_ios< wchar_t >	
std::basic_ostream< char >	
std::basic_ostream< wchar_t >	
std::basic_istream< char >	
std::basic_istream< wchar_t >	
std::basic_istringstream< char >	
std::basic_istringstream< wchar_t >	
std::basic_ofstream< char >	
std::basic_ofstream< wchar_t >	
std::basic_ostream< char >	
std::basic_ostream< wchar_t >	
std::basic_ostringstream< char >	
std::basic_ostringstream< wchar_t >	
std::basic_string< char >	
std::basic_string< wchar_t >	
std::basic_stringstream< char >	
std::basic_stringstream< wchar_t >	
<b>SIMCRS::BomAbstract</b>	<b>86</b>
<b>SIMCRS::DistributionManager</b>	<b>88</b>
<b>SIMCRS::FacBomAbstract</b>	<b>88</b>
<b>SIMCRS::FacServiceAbstract</b>	<b>91</b>
<b>SIMCRS::FacSimcrsServiceContext</b>	<b>92</b>
<b>SIMCRS::FacSupervisor</b>	<b>94</b>
<b>RootException</b>	<b>97</b>
<b>SIMCRS::AvailabilityRetrievalException</b>	<b>86</b>
<b>SIMCRS::BookingException</b>	<b>88</b>
<b>SIMCRS::ServiceAbstract</b>	<b>97</b>
<b>SIMCRS::SIMCRS_ServiceContext</b>	<b>104</b>
<b>SIMCRS::SIMCRS_Service</b>	<b>98</b>

## 19 Class Index

### 19.1 Class List

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

<b>SIMCRS::AvailabilityRetrievalException</b>	<b>86</b>
<b>SIMCRS::BomAbstract</b>	<b>86</b>
<b>SIMCRS::BookingException</b>	<b>88</b>

<a href="#">SIMCRS::DistributionManager</a>	
Command wrapping the travel distribution (CRS/GDS) process	88
<a href="#">SIMCRS::FacBomAbstract</a>	88
<a href="#">SIMCRS::FacServiceAbstract</a>	91
<a href="#">SIMCRS::FacSimcrsServiceContext</a>	92
<a href="#">SIMCRS::FacSupervisor</a>	94
<a href="#">RootException</a>	97
<a href="#">SIMCRS::ServiceAbstract</a>	97
<a href="#">SIMCRS::SIMCRS_Service</a>	98
<a href="#">SIMCRS::SIMCRS_ServiceContext</a>	
Class holding the context of the Simcrs services	104

## 20 File Index

### 20.1 File List

Here is a list of all files with brief descriptions:

<a href="#">simcrs/SIMCRS_Service.hpp</a>	144
<a href="#">simcrs/SIMCRS_Types.hpp</a>	146
<a href="#">simcrs/basic/BasConst.cpp</a>	106
<a href="#">simcrs/basic/BasConst_General.hpp</a>	106
<a href="#">simcrs/basic/BasConst_SIMCRS_Service.hpp</a>	106
<a href="#">simcrs/batches/simcrs.cpp</a>	109
<a href="#">simcrs/bom/BomAbstract.cpp</a>	114
<a href="#">simcrs/bom/BomAbstract.hpp</a>	115
<a href="#">simcrs/command/DistributionManager.cpp</a>	116
<a href="#">simcrs/command/DistributionManager.hpp</a>	118
<a href="#">simcrs/config/simcrs-paths.hpp</a>	120
<a href="#">simcrs/config/simcrs-paths.hpp.in</a>	122
<a href="#">simcrs/factory/FacBomAbstract.cpp</a>	123
<a href="#">simcrs/factory/FacBomAbstract.hpp</a>	124
<a href="#">simcrs/factory/FacServiceAbstract.cpp</a>	125
<a href="#">simcrs/factory/FacServiceAbstract.hpp</a>	125
<a href="#">simcrs/factory/FacSimcrsServiceContext.cpp</a>	126
<a href="#">simcrs/factory/FacSimcrsServiceContext.hpp</a>	127

<a href="#">simcrs/factory/FacSupervisor.cpp</a>	128
<a href="#">simcrs/factory/FacSupervisor.hpp</a>	129
<a href="#">simcrs/service/ServiceAbstract.cpp</a>	130
<a href="#">simcrs/service/ServiceAbstract.hpp</a>	131
<a href="#">simcrs/service/SIMCRS_Service.cpp</a>	132
<a href="#">simcrs/service/SIMCRS_ServiceContext.cpp</a>	140
<a href="#">simcrs/service/SIMCRS_ServiceContext.hpp</a>	142
<a href="#">test/simcrs/CRSTestSuite.cpp</a>	146

## 21 Namespace Documentation

### 21.1 AIRINV Namespace Reference

### 21.2 SIMCRS Namespace Reference

#### Classes

- class [BomAbstract](#)
- class [DistributionManager](#)  
*Command wrapping the travel distribution (CRS/GDS) process.*
- class [FacBomAbstract](#)
- class [FacServiceAbstract](#)
- class [FacSimcrsServiceContext](#)
- class [FacSupervisor](#)
- class [ServiceAbstract](#)
- class [SIMCRS\\_ServiceContext](#)  
*Class holding the context of the Simcrs services.*
- class [SIMCRS\\_Service](#)
- class [BookingException](#)
- class [AvailabilityRetrievalException](#)

#### Typedefs

- typedef std::string [CRSCode\\_T](#)
- typedef boost::shared\_ptr  
    < [SIMCRS\\_Service](#) > [SIMCRS\\_ServicePtr\\_T](#)

#### Variables

- const std::string [DEFAULT\\_CRS\\_CODE](#) = "1S"

#### 21.2.1 Typedef Documentation

##### 21.2.1.1 typedef std::string SIMCRS::CRSCode\_T

CRS code (identifier of the CRS; not actually used for now).

Definition at line 39 of file [SIMCRS\\_Types.hpp](#).

21.2.1.2 `typedef boost::shared_ptr<SIMCRS_Service> SIMCRS::SIMCRS_ServicePtr_T`

(Smart) Pointer on the SimCRS service handler.

Definition at line 44 of file [SIMCRS\\_Types.hpp](#).

## 21.2.2 Variable Documentation

21.2.2.1 `const std::string SIMCRS::DEFAULT_CRS_CODE = "1S"`

Default CRS code for the [SIMCRS\\_Service](#).

Definition at line 10 of file [BasConst.cpp](#).

## 21.3 stdair Namespace Reference

Forward declarations.

### 21.3.1 Detailed Description

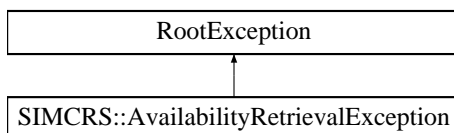
Forward declarations.

## 22 Class Documentation

### 22.1 SIMCRS::AvailabilityRetrievalException Class Reference

`#include <simcrs/SIMCRS_Types.hpp>`

Inheritance diagram for SIMCRS::AvailabilityRetrievalException:



#### 22.1.1 Detailed Description

Specific exception related to availability calculation.

Definition at line 31 of file [SIMCRS\\_Types.hpp](#).

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

- [simcrs/SIMCRS\\_Types.hpp](#)

### 22.2 SIMCRS::BomAbstract Class Reference

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

#### Public Member Functions

- virtual void [toStream](#) (std::ostream &ioOut) const =0

- virtual void [fromStream](#) (std::istream &ioln)=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](#)

#### 22.2.1 Detailed Description

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

Definition at line 14 of file [BomAbstract.hpp](#).

#### 22.2.2 Constructor & Destructor Documentation

##### 22.2.2.1 SIMCRS::BomAbstract::BomAbstract ( ) [inline],[protected]

Protected Default Constructor to ensure this class is abstract.

Definition at line 40 of file [BomAbstract.hpp](#).

##### 22.2.2.2 SIMCRS::BomAbstract::BomAbstract ( const BomAbstract & ) [inline],[protected]

Definition at line 41 of file [BomAbstract.hpp](#).

##### 22.2.2.3 virtual SIMCRS::BomAbstract::~~BomAbstract ( ) [inline],[protected],[virtual]

Destructor.

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

#### 22.2.3 Member Function Documentation

##### 22.2.3.1 virtual void SIMCRS::BomAbstract::toStream ( std::ostream & ioOut ) const [pure virtual]

Dump a Business Object into an output stream.

#### Parameters

<i>ostream&amp;</i>	the output stream.
---------------------	--------------------

##### 22.2.3.2 virtual void SIMCRS::BomAbstract::fromStream ( std::istream & ioln ) [pure virtual]

Read a Business Object from an input stream.

#### Parameters

<i>istream&amp;</i>	the input stream.
---------------------	-------------------

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

**22.2.3.3** `virtual std::string SIMCRS::BomAbstract::toString ( ) const [pure virtual]`

Get the serialised version of the Business Object.

**22.2.3.4** `virtual std::string SIMCRS::BomAbstract::describeKey ( ) const [pure virtual]`

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

**22.2.3.5** `virtual std::string SIMCRS::BomAbstract::describeShortKey ( ) const [pure virtual]`

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

## 22.2.4 Friends And Related Function Documentation

**22.2.4.1** `friend class FacBomAbstract [friend]`

Definition at line 15 of file [BomAbstract.hpp](#).

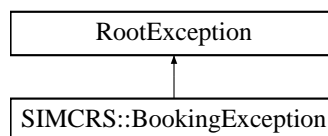
The documentation for this class was generated from the following file:

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

## 22.3 SIMCRS::BookingException Class Reference

```
#include <simcrs/SIMCRS_Types.hpp>
```

Inheritance diagram for SIMCRS::BookingException:



### 22.3.1 Detailed Description

Specific exception related to bookings made against the CRS.

Definition at line 25 of file [SIMCRS\\_Types.hpp](#).

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

- [simcrs/SIMCRS\\_Types.hpp](#)

## 22.4 SIMCRS::DistributionManager Class Reference

Command wrapping the travel distribution (CRS/GDS) process.

```
#include <simcrs/command/DistributionManager.hpp>
```

### Friends

- class [SIMCRS\\_Service](#)

### 22.4.1 Detailed Description

Command wrapping the travel distribution (CRS/GDS) process.

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

### 22.4.2 Friends And Related Function Documentation

#### 22.4.2.1 friend class SIMCRS\_Service [friend]

Definition at line 32 of file [DistributionManager.hpp](#).

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

- [simcrs/command/DistributionManager.hpp](#)
- [simcrs/command/DistributionManager.cpp](#)

## 22.5 SIMCRS::FacBomAbstract Class Reference

```
#include <simcrs/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](#)

### 22.5.1 Detailed Description

Base class for Factory layer.

Definition at line 17 of file [FacBomAbstract.hpp](#).



### 22.5.2 Member Typedef Documentation

#### 22.5.2.1 `typedef std::vector<BomAbstract*> SIMCRS::FacBomAbstract::BomPool_T`

Define the list (pool) of Bom objects.

Definition at line 22 of file [FacBomAbstract.hpp](#).

### 22.5.3 Constructor & Destructor Documentation

#### 22.5.3.1 `SIMCRS::FacBomAbstract::FacBomAbstract ( ) [inline], [protected]`

Default Constructor.

This constructor is protected to ensure the class is abstract.

Definition at line 41 of file [FacBomAbstract.hpp](#).

#### 22.5.3.2 `SIMCRS::FacBomAbstract::FacBomAbstract ( const FacBomAbstract & ) [inline], [protected]`

Definition at line 42 of file [FacBomAbstract.hpp](#).

#### 22.5.3.3 `SIMCRS::FacBomAbstract::~~FacBomAbstract ( ) [protected], [virtual]`

Destructor.

Definition at line 16 of file [FacBomAbstract.cpp](#).

### 22.5.4 Member Function Documentation

#### 22.5.4.1 `std::size_t SIMCRS::FacBomAbstract::getID ( const BomAbstract * iBomAbstract_ptr ) [static]`

Return the ID corresponding to the given object pointer.

Definition at line 35 of file [FacBomAbstract.cpp](#).

Referenced by [getID\(\)](#), and [getIDString\(\)](#).

#### 22.5.4.2 `std::size_t SIMCRS::FacBomAbstract::getID ( const BomAbstract & iBomAbstract ) [static]`

Return the ID corresponding to the given object reference.

Definition at line 43 of file [FacBomAbstract.cpp](#).

References [getID\(\)](#).

#### 22.5.4.3 `std::string SIMCRS::FacBomAbstract::getIDString ( const BomAbstract * iBomAbstract_ptr ) [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\(\)](#).

#### 22.5.4.4 `std::string SIMCRS::FacBomAbstract::getIDString ( const BomAbstract & iBomAbstract ) [static]`

Return the ID, as a string, corresponding to the given object reference.

Definition at line 56 of file [FacBomAbstract.cpp](#).

References [getIDString\(\)](#).

## 22.5.5 Friends And Related Function Documentation

## 22.5.5.1 friend class FacSupervisor [friend]

Definition at line 18 of file [FacBomAbstract.hpp](#).

## 22.5.6 Member Data Documentation

## 22.5.6.1 BomPool\_T SIMCRS::FacBomAbstract::pool [protected]

List of instantiated Business Objects

Definition at line 53 of file [FacBomAbstract.hpp](#).

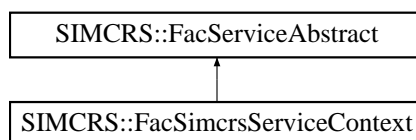
The documentation for this class was generated from the following files:

- [simcrs/factory/FacBomAbstract.hpp](#)
- [simcrs/factory/FacBomAbstract.cpp](#)

## 22.6 SIMCRS::FacServiceAbstract Class Reference

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

Inheritance diagram for SIMCRS::FacServiceAbstract:



## Public Types

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

## Public Member Functions

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

## Protected Member Functions

- [FacServiceAbstract](#) ()

## Protected Attributes

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

## 22.6.1 Detailed Description

Base class for the (Service) Factory layer.

Definition at line 16 of file [FacServiceAbstract.hpp](#).

### 22.6.2 Member Typedef Documentation

#### 22.6.2.1 `typedef std::vector<ServiceAbstract*> SIMCRS::FacServiceAbstract::ServicePool_T`

Define the list (pool) of Service objects.

Definition at line 20 of file [FacServiceAbstract.hpp](#).

### 22.6.3 Constructor & Destructor Documentation

#### 22.6.3.1 `SIMCRS::FacServiceAbstract::~~FacServiceAbstract ( ) [virtual]`

Destructor.

Definition at line 13 of file [FacServiceAbstract.cpp](#).

References [clean\(\)](#).

#### 22.6.3.2 `SIMCRS::FacServiceAbstract::FacServiceAbstract ( ) [inline],[protected]`

Default Constructor.

This constructor is protected to ensure the class is abstract.

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

### 22.6.4 Member Function Documentation

#### 22.6.4.1 `void SIMCRS::FacServiceAbstract::clean ( )`

Destroyed all the object instantiated by this factory.

Definition at line 18 of file [FacServiceAbstract.cpp](#).

References [\\_pool](#).

Referenced by [~FacServiceAbstract\(\)](#).

### 22.6.5 Member Data Documentation

#### 22.6.5.1 `ServicePool_T SIMCRS::FacServiceAbstract::_pool [protected]`

List of instantiated Business Objects

Definition at line 34 of file [FacServiceAbstract.hpp](#).

Referenced by [clean\(\)](#), and [SIMCRS::FacSimcrsServiceContext::create\(\)](#).

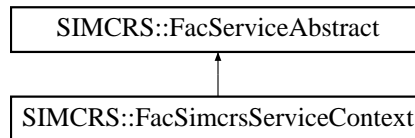
The documentation for this class was generated from the following files:

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

## 22.7 SIMCRS::FacSimcrsServiceContext Class Reference

```
#include <simcrs/factory/FacSimcrsServiceContext.hpp>
```

Inheritance diagram for SIMCRS::FacSimcrsServiceContext:



### Public Types

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

### Public Member Functions

- [~FacSimcrsServiceContext](#) ()
- [SIMCRS\\_ServiceContext](#) & [create](#) (const std::string &iTravelDatabaseName)
- void [clean](#) ()

### Static Public Member Functions

- static [FacSimcrsServiceContext](#) & [instance](#) ()

### Protected Member Functions

- [FacSimcrsServiceContext](#) ()

### Protected Attributes

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

#### 22.7.1 Detailed Description

Factory for Bucket.

Definition at line 18 of file [FacSimcrsServiceContext.hpp](#).

#### 22.7.2 Member Typedef Documentation

**22.7.2.1** typedef std::vector<[ServiceAbstract](#)\*> [SIMCRS::FacServiceAbstract::ServicePool\\_T](#)  
[inherited]

Define the list (pool) of Service objects.

Definition at line 20 of file [FacServiceAbstract.hpp](#).

#### 22.7.3 Constructor & Destructor Documentation

**22.7.3.1** [SIMCRS::FacSimcrsServiceContext::~~FacSimcrsServiceContext](#) ( )

Destructor.

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

Definition at line 16 of file [FacSimcrsServiceContext.cpp](#).

### 22.7.3.2 SIMCRS::FacSimcrsServiceContext::FacSimcrsServiceContext ( ) [inline],[protected]

Default Constructor.

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

Definition at line 42 of file [FacSimcrsServiceContext.hpp](#).

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

## 22.7.4 Member Function Documentation

### 22.7.4.1 FacSimcrsServiceContext & SIMCRS::FacSimcrsServiceContext::instance ( ) [static]

Provide the unique instance.

The singleton is instantiated when first used

Returns

[FacSimcrsServiceContext&](#)

Definition at line 21 of file [FacSimcrsServiceContext.cpp](#).

References [FacSimcrsServiceContext\(\)](#).

### 22.7.4.2 SIMCRS\_ServiceContext & SIMCRS::FacSimcrsServiceContext::create ( const std::string & *iTravelDatabaseName* )

Create a new [SIMCRS\\_ServiceContext](#) object.

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

Returns

[SIMCRS\\_ServiceContext&](#) The newly created object.

Definition at line 34 of file [FacSimcrsServiceContext.cpp](#).

References [SIMCRS::FacServiceAbstract::\\_pool](#).

### 22.7.4.3 void SIMCRS::FacServiceAbstract::clean ( ) [inherited]

Destroyed all the object instantiated by this factory.

Definition at line 18 of file [FacServiceAbstract.cpp](#).

References [SIMCRS::FacServiceAbstract::\\_pool](#).

Referenced by [SIMCRS::FacServiceAbstract::~~FacServiceAbstract\(\)](#).

## 22.7.5 Member Data Documentation

### 22.7.5.1 ServicePool\_T SIMCRS::FacServiceAbstract::\_pool [protected],[inherited]

List of instantiated Business Objects

Definition at line 34 of file [FacServiceAbstract.hpp](#).

Referenced by [SIMCRS::FacServiceAbstract::clean\(\)](#), and [create\(\)](#).

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

- [simcrs/factory/FacSimcrsServiceContext.hpp](#)
- [simcrs/factory/FacSimcrsServiceContext.cpp](#)

## 22.8 SIMCRS::FacSupervisor Class Reference

```
#include <simcrs/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](#) &)

#### 22.8.1 Detailed Description

Singleton class to register and clean all Factories.

Definition at line 17 of file [FacSupervisor.hpp](#).

#### 22.8.2 Member Typedef Documentation

##### 22.8.2.1 typedef std::vector<[FacBomAbstract](#)\*> [SIMCRS::FacSupervisor::BomFactoryPool\\_T](#)

Define the pool (list) of factories.

Definition at line 21 of file [FacSupervisor.hpp](#).

##### 22.8.2.2 typedef std::vector<[FacServiceAbstract](#)\*> [SIMCRS::FacSupervisor::ServiceFactoryPool\\_T](#)

Definition at line 22 of file [FacSupervisor.hpp](#).

#### 22.8.3 Constructor & Destructor Documentation

##### 22.8.3.1 [SIMCRS::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\(\)](#).

#### 22.8.3.2 SIMCRS::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\(\)](#).

#### 22.8.3.3 SIMCRS::FacSupervisor::FacSupervisor ( const FacSupervisor & ) [inline], [protected]

Definition at line 66 of file [FacSupervisor.hpp](#).

### 22.8.4 Member Function Documentation

#### 22.8.4.1 FacSupervisor & SIMCRS::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\(\)](#).

#### 22.8.4.2 void SIMCRS::FacSupervisor::registerBomFactory ( FacBomAbstract \* ioFacBomAbstract\_ptr )

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

<i>FacAbstract&amp;</i>	the concrete Factory to register.
-------------------------	-----------------------------------

Definition at line 30 of file [FacSupervisor.cpp](#).

#### 22.8.4.3 void SIMCRS::FacSupervisor::registerServiceFactory ( FacServiceAbstract \* ioFacServiceAbstract\_ptr )

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

<i>FacService-Abstract&amp;</i>	the concrete Factory to register.
---------------------------------	-----------------------------------

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

#### 22.8.4.4 void SIMCRS::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\(\)](#).

#### 22.8.4.5 void SIMCRS::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\(\)](#).

#### 22.8.4.6 void SIMCRS::FacSupervisor::cleanFactory ( ) [static]

Clean the static instance.

The singleton is deleted.

Definition at line 75 of file [FacSupervisor.cpp](#).

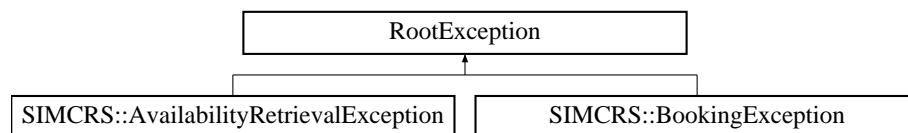
References [cleanBomLayer\(\)](#), and [cleanServiceLayer\(\)](#).

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

- [simcrs/factory/FacSupervisor.hpp](#)
- [simcrs/factory/FacSupervisor.cpp](#)

## 22.9 RootException Class Reference

Inheritance diagram for RootException:



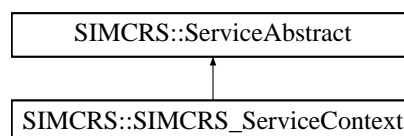
The documentation for this class was generated from the following file:

- [simcrs/SIMCRS\\_Types.hpp](#)

## 22.10 SIMCRS::ServiceAbstract Class Reference

```
#include <simcrs/service/ServiceAbstract.hpp>
```

Inheritance diagram for SIMCRS::ServiceAbstract:



### Public Member Functions

- virtual [~ServiceAbstract](#) ( )
- virtual void [toStream](#) (std::ostream &ioOut) const
- virtual void [fromStream](#) (std::istream &ioIn)



## Protected Member Functions

- [ServiceAbstract](#) ()

## 22.10.1 Detailed Description

Base class for the Service layer.

Definition at line 14 of file [ServiceAbstract.hpp](#).

## 22.10.2 Constructor &amp; Destructor Documentation

22.10.2.1 `virtual SIMCRS::ServiceAbstract::~~ServiceAbstract ( ) [inline],[virtual]`

Destructor.

Definition at line 18 of file [ServiceAbstract.hpp](#).

22.10.2.2 `SIMCRS::ServiceAbstract::ServiceAbstract ( ) [inline],[protected]`

Protected Default Constructor to ensure this class is abstract.

Definition at line 30 of file [ServiceAbstract.hpp](#).

## 22.10.3 Member Function Documentation

22.10.3.1 `virtual void SIMCRS::ServiceAbstract::toStream ( std::ostream & ioOut ) const [inline],[virtual]`

Dump a Business Object into an output stream.

## Parameters

<i>ostream&amp;</i>	the output stream.
---------------------	--------------------

Definition at line 22 of file [ServiceAbstract.hpp](#).

22.10.3.2 `virtual void SIMCRS::ServiceAbstract::fromStream ( std::istream & ioIn ) [inline],[virtual]`

Read a Business Object from an input stream.

## Parameters

<i>istream&amp;</i>	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:

- [simcrs/service/ServiceAbstract.hpp](#)

## 22.11 SIMCRS::SIMCRS\_Service Class Reference

```
#include <simcrs/SIMCRS_Service.hpp>
```

## Public Member Functions

- [SIMCRS\\_Service](#) (const stdair::BasLogParams &, const stdair::BasDBParams &, const [CRSCode\\_T](#) &)

- [SIMCRS\\_Service](#) (const stdair::BasLogParams &, const [CRSCode\\_T](#) &)
- [SIMCRS\\_Service](#) (stdair::STDAIR\_ServicePtr\_T, const [CRSCode\\_T](#) &)
- void [parseAndLoad](#) (const stdair::Filename\_T &iScheduleInputFilename, const stdair::Filename\_T &iOD-InputFilename, const AIRRAC::YieldFilePath &iYieldInputFilepath, const SIMFQT::FareFilePath &iFareInput-Filepath)
- void [initSnapshotAndRMEvents](#) (const stdair::Date\_T &iStartDate, const stdair::Date\_T &iEndDate)
- [~SIMCRS\\_Service](#) ()
- stdair::TravelSolutionList\_T [calculateSegmentPathList](#) (const stdair::BookingRequestStruct &)
- void [fareQuote](#) (const stdair::BookingRequestStruct &, stdair::TravelSolutionList\_T &)
- void [calculateAvailability](#) (stdair::TravelSolutionList\_T &, const stdair::PartnershipTechnique &)
- bool [sell](#) (const stdair::TravelSolutionStruct &, const stdair::PartySize\_T &)
- void [takeSnapshots](#) (const stdair::SnapshotStruct &)
- bool [playCancellation](#) (const stdair::CancellationStruct &)
- void [optimise](#) (const stdair::RMEventStruct &, const stdair::ForecastingMethod &, const stdair::Partnership-Technique &)
- void [buildSampleBom](#) ()
- void [buildSampleTravelSolutions](#) (stdair::TravelSolutionList\_T &)
- stdair::BookingRequestStruct [buildSampleBookingRequest](#) (const bool isForCRS=false)
- std::string [jsonExport](#) (const stdair::AirlineCode\_T &, const stdair::FlightNumber\_T &, const stdair::Date\_T &iDepartureDate) const
- std::string [csvDisplay](#) () const
- std::string [csvDisplay](#) (const stdair::TravelSolutionList\_T &) const

### 22.11.1 Detailed Description

Interface for the [SIMCRS](#) Services.

Definition at line 39 of file [SIMCRS\\_Service.hpp](#).

### 22.11.2 Constructor & Destructor Documentation

#### 22.11.2.1 SIMCRS::SIMCRS\_Service::SIMCRS\_Service ( const stdair::BasLogParams & *iLogParams*, const stdair::BasDBParams & *iDBParams*, const [CRSCode\\_T](#) & *iCRSCode* )

Constructor.

The init() 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

<i>const</i>	stdair::BasLogParams& Parameters for the output log stream.
<i>const</i>	stdair::BasDBParams& Parameters for the database access.
<i>const</i>	<a href="#">CRSCode_T</a> & Code of the owner of the distribution system.

Definition at line 78 of file [SIMCRS\\_Service.cpp](#).

#### 22.11.2.2 SIMCRS::SIMCRS\_Service::SIMCRS\_Service ( const stdair::BasLogParams & *iLogParams*, const [CRSCode\\_T](#) & *iCRSCode* )

Constructor.

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

Moreover, a reference on an output stream is given, so that log outputs can be directed onto that stream.

## Parameters

<i>const</i>	stdair::BasLogParams& Parameters for the output log stream.
<i>const</i>	CRSCode_T& Code of the owner of the distribution system.

Definition at line 48 of file [SIMCRS\\_Service.cpp](#).

#### 22.11.2.3 SIMCRS::SIMCRS\_Service::SIMCRS\_Service ( stdair::STDAIR\_ServicePtr\_T ioSTDAIR\_Service\_ptr, const CRSCode\_T & iCRSCode )

Constructor.

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

## Parameters

<i>stdair::STDAIR_ServicePtr_T</i>	Reference on the STDAIR service.
<i>const</i>	CRSCode_T& Code of the owner of the distribution system.

Definition at line 110 of file [SIMCRS\\_Service.cpp](#).

#### 22.11.2.4 SIMCRS::SIMCRS\_Service::~SIMCRS\_Service ( )

Destructor.

Definition at line 136 of file [SIMCRS\\_Service.cpp](#).

### 22.11.3 Member Function Documentation

#### 22.11.3.1 void SIMCRS::SIMCRS\_Service::parseAndLoad ( const stdair::Filename\_T & iScheduleInputFilename, const stdair::Filename\_T & iODInputFilename, const AIRRAC::YieldFilePath & iYieldInputFilepath, const SIMFQT::FareFilePath & iFareInputFilepath )

Parse the schedule, O&D, fare and yield input files.

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

## Parameters

<i>const</i>	stdair::Filename_T& Filename of the input schedule file.
<i>const</i>	stdair::Filename_T& Filename of the input O&D file.
<i>const</i>	AIRRAC::YieldFilePath& Filename of the input yield file.
<i>const</i>	SIMFQT::FareFilePath& Filename of the input fare file.

Definition at line 288 of file [SIMCRS\\_Service.cpp](#).

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

#### 22.11.3.2 void SIMCRS::SIMCRS\_Service::initSnapshotAndRMEvents ( const stdair::Date\_T & iStartDate, const stdair::Date\_T & iEndDate )

Initialise the snapshot and RM events for the inventories.

## Parameters

<i>const</i>	stdair::Date_T& Start date of the simulation.
<i>const</i>	stdair::Date_T& End date of the simulation.

Definition at line 457 of file [SIMCRS\\_Service.cpp](#).

22.11.3.3 `stdair::TravelSolutionList_T SIMCRS::SIMCRS_Service::calculateSegmentPathList ( const stdair::BookingRequestStruct & iBookingRequest )`

Construct the list of travel solutions corresponding to the booking request.

Definition at line 516 of file [SIMCRS\\_Service.cpp](#).

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

22.11.3.4 `void SIMCRS::SIMCRS_Service::fareQuote ( const stdair::BookingRequestStruct & iBookingRequest, stdair::TravelSolutionList_T & ioTravelSolutionList )`

Calculate the fare of each travel solutions in the list.

Definition at line 552 of file [SIMCRS\\_Service.cpp](#).

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

22.11.3.5 `void SIMCRS::SIMCRS_Service::calculateAvailability ( stdair::TravelSolutionList_T & ioTravelSolutionList, const stdair::PartnershipTechnique & iPartnershipTechnique )`

Compute the availability for each travel solution in the list.

Definition at line 583 of file [SIMCRS\\_Service.cpp](#).

22.11.3.6 `bool SIMCRS::SIMCRS_Service::sell ( const stdair::TravelSolutionStruct & iTravelSolution, const stdair::PartySize_T & iPartySize )`

Register a booking.

Definition at line 618 of file [SIMCRS\\_Service.cpp](#).

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

22.11.3.7 `void SIMCRS::SIMCRS_Service::takeSnapshots ( const stdair::SnapshotStruct & iSnapshot )`

Take inventory snapshots.

Definition at line 704 of file [SIMCRS\\_Service.cpp](#).

22.11.3.8 `bool SIMCRS::SIMCRS_Service::playCancellation ( const stdair::CancellationStruct & iCancellation )`

Play cancellation.

Definition at line 665 of file [SIMCRS\\_Service.cpp](#).

22.11.3.9 `void SIMCRS::SIMCRS_Service::optimise ( const stdair::RMEEventStruct & iRMEEvent, const stdair::ForecastingMethod & iForecastingMethod, const stdair::PartnershipTechnique & iPartnershipTechnique )`

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

Definition at line 723 of file [SIMCRS\\_Service.cpp](#).

22.11.3.10 `void SIMCRS::SIMCRS_Service::buildSampleBom ( )`

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

As for now, the BOM sample tree is the one built by the AirInv component.

See Also

AIRINV::AIRINV\_Master\_Service and stdair::CmdBomManager for more details.

Definition at line 326 of file [SIMCRS\\_Service.cpp](#).

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

22.11.3.11 void SIMCRS::SIMCRS\_Service::buildSampleTravelSolutions ( stdair::TravelSolutionList\_T & ioTravelSolutionList )

Build a sample list of travel solutions.

As of now (March 2011), that list is made of the following travel solutions:

- BA9
- LHR-SYD
- 2011-06-10
- Q
- WTP: 900
- Change fee: 20; Non refundable; Saturday night stay

#### See Also

stdair::CmdBomManager for more details.

#### Parameters

<i>TravelSolution-List_T&amp;</i>	Sample list of travel solution structures. It should be given empty. It is altered with the returned sample.
-----------------------------------	--

Definition at line 394 of file [SIMCRS\\_Service.cpp](#).

22.11.3.12 stdair::BookingRequestStruct SIMCRS::SIMCRS\_Service::buildSampleBookingRequest ( const bool isForCRS = false )

Build a sample booking request structure.

As of now (March 2011), the sample booking request is made of the following parameters:

- Return trip (inbound): LHR-SYD (POS: LHR, Channel: DN),
- Departing 10-JUN-2011 around 8:00, staying 7 days
- Requested on 15-MAY-2011 at 10:00
- Economy cabin, 3 persons, FF member
- WTP: 1000.0 EUR
- Dis-utility: 100.0 EUR/hour

As of now (March 2011), the CRS-related booking request is made of the following parameters:

- Return trip (inbound): SIN-BKK (POS: SIN, Channel: IN),
- Departing 30-JAN-2010 around 10:00, staying 7 days
- Requested on 22-JAN-2010 at 10:00
- Economy cabin, 3 persons, FF member
- WTP: 1000.0 EUR
- Dis-utility: 100.0 EUR/hour

## See Also

stdair::CmdBomManager for more details.

## Parameters

<i>const</i>	bool isForCRS Whether the sample booking request is for CRS.
--------------	--

## Returns

BookingRequestStruct& Sample booking request structure.

Definition at line 414 of file [SIMCRS\\_Service.cpp](#).

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

**22.11.3.13** `std::string SIMCRS::SIMCRS_Service::jsonExport ( const stdair::AirlineCode_T & iAirlineCode, const stdair::FlightNumber_T & iFlightNumber, 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

<i>const</i>	stdair::AirlineCode_T& Airline code of the flight to dump.
<i>const</i>	stdair::FlightNumber_T& Flight number of the flight to dump.
<i>const</i>	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 434 of file [SIMCRS\\_Service.cpp](#).

**22.11.3.14** `std::string SIMCRS::SIMCRS_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 476 of file [SIMCRS\\_Service.cpp](#).

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

**22.11.3.15** `std::string SIMCRS::SIMCRS_Service::csvDisplay ( const stdair::TravelSolutionList_T & ioTravelSolutionList ) const`

Display (dump in the returned string) the full list of travel solution structures.

## Returns

std::string Output string in which the list of travel solutions is logged/dumped.

Definition at line 496 of file [SIMCRS\\_Service.cpp](#).

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

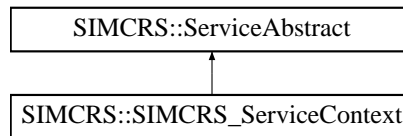
- [simcrs/SIMCRS\\_Service.hpp](#)
- [simcrs/service/SIMCRS\\_Service.cpp](#)

## 22.12 SIMCRS::SIMCRS\_ServiceContext Class Reference

Class holding the context of the Simcrs services.

```
#include <simcrs/service/SIMCRS_ServiceContext.hpp>
```

Inheritance diagram for SIMCRS::SIMCRS\_ServiceContext:



### Public Member Functions

- virtual void [toStream](#) (std::ostream &ioOut) const
- virtual void [fromStream](#) (std::istream &ioln)

### Friends

- class [SIMCRS\\_Service](#)
- class [FacSimcrsServiceContext](#)

### 22.12.1 Detailed Description

Class holding the context of the Simcrs services.

Definition at line 30 of file [SIMCRS\\_ServiceContext.hpp](#).

### 22.12.2 Member Function Documentation

**22.12.2.1** virtual void SIMCRS::ServiceAbstract::toStream ( std::ostream & *ioOut* ) const [inline], [virtual], [inherited]

Dump a Business Object into an output stream.

#### Parameters

<i>ostream&amp;</i>	the output stream.
---------------------	--------------------

Definition at line 22 of file [ServiceAbstract.hpp](#).

**22.12.2.2** virtual void SIMCRS::ServiceAbstract::fromStream ( std::istream & *ioln* ) [inline], [virtual], [inherited]

Read a Business Object from an input stream.

#### Parameters

<i>istream&amp;</i>	the input stream.
---------------------	-------------------

Definition at line 26 of file [ServiceAbstract.hpp](#).

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

### 22.12.3 Friends And Related Function Documentation

#### 22.12.3.1 friend class `SIMCRS_Service` [`friend`]

The `SIMCRS_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 36 of file `SIMCRS_ServiceContext.hpp`.

#### 22.12.3.2 friend class `FacSimcrsServiceContext` [`friend`]

Definition at line 37 of file `SIMCRS_ServiceContext.hpp`.

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

- `simcrs/service/SIMCRS_ServiceContext.hpp`
- `simcrs/service/SIMCRS_ServiceContext.cpp`

## 23 File Documentation

### 23.1 `doc/local/authors.doc` File Reference

### 23.2 `doc/local/codingrules.doc` File Reference

### 23.3 `doc/local/copyright.doc` File Reference

### 23.4 `doc/local/documentation.doc` File Reference

### 23.5 `doc/local/features.doc` File Reference

### 23.6 `doc/local/help_wanted.doc` File Reference

### 23.7 `doc/local/howto_release.doc` File Reference

### 23.8 `doc/local/index.doc` File Reference

### 23.9 `doc/local/installation.doc` File Reference

### 23.10 `doc/local/linking.doc` File Reference

### 23.11 `doc/local/test.doc` File Reference

### 23.12 `doc/local/users_guide.doc` File Reference

### 23.13 `doc/local/verification.doc` File Reference

### 23.14 `doc/tutorial/tutorial.doc` File Reference

### 23.15 `simcrs/basic/BasConst.cpp` File Reference

```
#include <simcrs/basic/BasConst_General.hpp>
#include <simcrs/basic/BasConst_SIMCRS_Service.hpp>
```



## Namespaces

- namespace [SIMCRS](#)

## Variables

- const std::string [SIMCRS::DEFAULT\\_CRS\\_CODE](#) = "1S"

## 23.16 BasConst.cpp

```

00001 // //////////////////////////////////////
00002 // Import section
00003 // //////////////////////////////////////
00004 #include <simcrs/basic/BasConst_General.hpp>
00005 #include <simcrs/basic/BasConst_SIMCRS_Service.hpp>
00006 >
00007 namespace SIMCRS {
00008
00010     const std::string DEFAULT_CRS_CODE = "1S";
00011
00012 }
```

## 23.17 simcrs/basic/BasConst\_General.hpp File Reference

## Namespaces

- namespace [SIMCRS](#)

## 23.18 BasConst\_General.hpp

```

00001 #ifndef __SIMCRS_BAS_BASCONST_GENERAL_HPP
00002 #define __SIMCRS_BAS_BASCONST_GENERAL_HPP
00003
00004 // //////////////////////////////////////
00005 // Import section
00006 // //////////////////////////////////////
00007
00008 namespace SIMCRS {
00009
00010 }
00011 #endif // __SIMCRS_BAS_BASCONST_GENERAL_HPP
```

## 23.19 simcrs/basic/BasConst\_SIMCRS\_Service.hpp File Reference

```
#include <string>
```

## Namespaces

- namespace [SIMCRS](#)

## 23.20 BasConst\_SIMCRS\_Service.hpp

```

00001 #ifndef __SIMCRS_BAS_BASCONST_SIMCRS_SERVICE_HPP
00002 #define __SIMCRS_BAS_BASCONST_SIMCRS_SERVICE_HPP
00003
00004 // //////////////////////////////////////
00005 // Import section
00006 // //////////////////////////////////////
00007 #include <string>
00008
00009 namespace SIMCRS {
```

```

00010
00012     extern const std::string DEFAULT_CRS_CODE;
00013
00014 }
00015 #endif // __SIMCRS_BAS_BASCONST_SIMCRS_SERVICE_HPP

```

## 23.21 simcrs/batches/simcrs.cpp File Reference

```

#include <sstream>
#include <fstream>
#include <string>
#include <boost/program_options.hpp>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/basic/BasLogParams.hpp>
#include <stdair/basic/BasDBParams.hpp>
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/bom/TravelSolutionStruct.hpp>
#include <stdair/bom/BookingRequestStruct.hpp>
#include <stdair/service/Logger.hpp>
#include <simfqt/SIMFQT_Types.hpp>
#include <simcrs/SIMCRS_Service.hpp>
#include <simcrs/config/simcrs-paths.hpp>

```

### Functions

- const std::string [K\\_SIMCRS\\_DEFAULT\\_LOG\\_FILENAME](#) ("simcrs.log")
- const std::string [K\\_SIMCRS\\_DEFAULT\\_SCHEDULE\\_INPUT\\_FILENAME](#) (STDAIR\_SAMPLE\_DIR"/schedule01.-csv")
- const std::string [K\\_SIMCRS\\_DEFAULT\\_OND\\_INPUT\\_FILENAME](#) (STDAIR\_SAMPLE\_DIR"/ond01.csv")
- const std::string [K\\_SIMCRS\\_DEFAULT\\_YIELD\\_INPUT\\_FILENAME](#) (STDAIR\_SAMPLE\_DIR"/yieldstore01.-csv")
- const std::string [K\\_SIMCRS\\_DEFAULT\\_FARE\\_INPUT\\_FILENAME](#) (STDAIR\_SAMPLE\_DIR"/fare01.csv")
- const std::string [K\\_SIMCRS\\_DEFAULT\\_DB\\_USER](#) ("dsim")
- const std::string [K\\_SIMCRS\\_DEFAULT\\_DB\\_PASSWD](#) ("dsim")
- const std::string [K\\_SIMCRS\\_DEFAULT\\_DB\\_DBNAME](#) ("sim\_dsim")
- const std::string [K\\_SIMCRS\\_DEFAULT\\_DB\\_HOST](#) ("localhost")
- const std::string [K\\_SIMCRS\\_DEFAULT\\_DB\\_PORT](#) ("3306")
- template<class T >  
std::ostream & [operator<<](#) (std::ostream &os, const std::vector< T > &v)
- int [readConfiguration](#) (int argc, char \*argv[], bool &iolsBuiltin, stdair::Filename\_T &ioScheduleInputFilename, stdair::Filename\_T &ioOnDInputFilename, stdair::Filename\_T &ioYieldInputFilename, stdair::Filename\_T &ioFareInputFilename, stdair::Filename\_T &ioLogFilename, std::string &ioDBUser, std::string &ioDBPasswd, std::string &ioDBHost, std::string &ioDBPort, std::string &ioDBDBName)
- int [main](#) (int argc, char \*argv[])

### Variables

- const bool [K\\_SIMCRS\\_DEFAULT\\_BUILT\\_IN\\_INPUT](#) = false
- const int [K\\_SIMCRS\\_EARLY\\_RETURN\\_STATUS](#) = 99

#### 23.21.1 Function Documentation

##### 23.21.1.1 const std::string K\_SIMCRS\_DEFAULT\_LOG\_FILENAME ( "simcrs.log" )

Default name and location for the log file.

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

23.21.1.2 `const std::string K_SIMCRS_DEFAULT_SCHEDULE_INPUT_FILENAME ( STDAIR_SAMPLE_DIR"/schedule01.csv"`  
`)`

Default name and location for the (CSV) schedule input file.

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

23.21.1.3 `const std::string K_SIMCRS_DEFAULT_OND_INPUT_FILENAME ( STDAIR_SAMPLE_DIR"/ond01.csv" )`

Default name and location for the (CSV) O&D input file.

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

23.21.1.4 `const std::string K_SIMCRS_DEFAULT_YIELD_INPUT_FILENAME ( STDAIR_SAMPLE_DIR"/yieldstore01.csv" )`

Default name and location for the (CSV) yield input file.

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

23.21.1.5 `const std::string K_SIMCRS_DEFAULT_FARE_INPUT_FILENAME ( STDAIR_SAMPLE_DIR"/fare01.csv" )`

Default name and location for the (CSV) fare input file.

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

23.21.1.6 `const std::string K_SIMCRS_DEFAULT_DB_USER ( "dsim" )`

Default name and location for the MySQL database.

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

23.21.1.7 `const std::string K_SIMCRS_DEFAULT_DB_PASSWD ( "dsim" )`

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

23.21.1.8 `const std::string K_SIMCRS_DEFAULT_DB_DBNAME ( "sim.dsim" )`

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

23.21.1.9 `const std::string K_SIMCRS_DEFAULT_DB_HOST ( "localhost" )`

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

23.21.1.10 `const std::string K_SIMCRS_DEFAULT_DB_PORT ( "3306" )`

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

23.21.1.11 `template<class T> std::ostream& operator<< ( std::ostream & os, const std::vector< T > & v )`

Definition at line 69 of file [simcrs.cpp](#).

23.21.1.12 `int readConfiguration ( int argc, char * argv[], bool & iolsBuiltin, stdair::Filename_T & ioScheduleInputFilename,  
 stdair::Filename_T & ioOnDInputFilename, stdair::Filename_T & ioYieldInputFilename, stdair::Filename_T &  
 ioFareInputFilename, stdair::Filename_T & ioLogFilename, std::string & ioDBUser, std::string & ioDBPasswd,  
 std::string & ioDBHost, std::string & ioDBPort, std::string & ioDBDBName )`

Read and parse the command line options.

Definition at line 79 of file [simcrs.cpp](#).

References [K\\_SIMCRS\\_DEFAULT\\_BUILT\\_IN\\_INPUT](#), [K\\_SIMCRS\\_DEFAULT\\_DB\\_DBNAME\(\)](#), [K\\_SIMCRS\\_DEFAULT\\_DB\\_HOST\(\)](#), [K\\_SIMCRS\\_DEFAULT\\_DB\\_PASSWD\(\)](#), [K\\_SIMCRS\\_DEFAULT\\_DB\\_PORT\(\)](#), [K\\_SIMCRS\\_DEFAULT\\_DB\\_USER\(\)](#), [K\\_SIMCRS\\_DEFAULT\\_FARE\\_INPUT\\_FILENAME\(\)](#), [K\\_SIMCRS\\_DEFAULT\\_LOG\\_FILENAME\(\)](#), [K\\_SIMCRS\\_DEFAULT\\_OND\\_INPUT\\_FILENAME\(\)](#), [K\\_SIMCRS\\_DEFAULT\\_SCHEDULE\\_INPUT\\_FILENAME\(\)](#), [K\\_SIMCRS\\_DEFAULT\\_YIELD\\_INPUT\\_FILENAME\(\)](#), [K\\_SIMCRS\\_EARLY\\_RETURN\\_STATUS](#), [PACK-](#)

[AGE\\_NAME](#), [PACKAGE\\_VERSION](#), and [PREFIXDIR](#).

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

23.21.1.13 `int main ( int argc, char * argv[] )`

Definition at line 273 of file [simcrs.cpp](#).

References [SIMCRS::SIMCRS\\_Service::buildSampleBom\(\)](#), [SIMCRS::SIMCRS\\_Service::buildSampleBookingRequest\(\)](#), [SIMCRS::SIMCRS\\_Service::calculateSegmentPathList\(\)](#), [SIMCRS::SIMCRS\\_Service::csvDisplay\(\)](#), [SIMCRS::SIMCRS\\_Service::fareQuote\(\)](#), [K\\_SIMCRS\\_EARLY\\_RETURN\\_STATUS](#), [SIMCRS::SIMCRS\\_Service::parseAndLoad\(\)](#), [readConfiguration\(\)](#), and [SIMCRS::SIMCRS\\_Service::sell\(\)](#).

## 23.21.2 Variable Documentation

23.21.2.1 `const bool K_SIMCRS_DEFAULT_BUILT_IN_INPUT = false`

Default for the BOM tree building. The BOM tree can either be built-in or provided by an input file. That latter must then be given with input file options (-s, -o, -f, -y).

Definition at line 56 of file [simcrs.cpp](#).

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

23.21.2.2 `const int K_SIMCRS_EARLY_RETURN_STATUS = 99`

Early return status (so that it can be differentiated from an error).

Definition at line 76 of file [simcrs.cpp](#).

Referenced by [main\(\)](#), and [readConfiguration\(\)](#).

## 23.22 simcrs.cpp

```
00001 // STL
00002 #include <sstream>
00003 #include <fstream>
00004 #include <string>
00005 // Boost (Extended STL)
00006 #include <boost/program_options.hpp>
00007 // StdAir
00008 #include <stdair/stdair_basic_types.hpp>
00009 #include <stdair/basic/BasLogParams.hpp>
00010 #include <stdair/basic/BasDBParams.hpp>
00011 #include <stdair/basic/BasFileMgr.hpp>
00012 #include <stdair/bom/TravelSolutionStruct.hpp>
00013 #include <stdair/bom/BookingRequestStruct.hpp>
00014 #include <stdair/service/Logger.hpp>
00015 // SimFQT
00016 #include <simfqt/SIMFQT_Types.hpp>
00017 // SimCRS
00018 #include <simcrs/SIMCRS_Service.hpp>
00019 #include <simcrs/config/simcrs-paths.hpp>
00020
00021 // ////////// Constants //////////
00025 const std::string K_SIMCRS_DEFAULT_LOG_FILENAME ( "
    simcrs.log" );
00026
00030 const std::string K_SIMCRS_DEFAULT_SCHEDULE_INPUT_FILENAME
    (STDAIR_SAMPLE_DIR
    "/schedule01.csv" );
00031
00032
00036 const std::string K_SIMCRS_DEFAULT_OND_INPUT_FILENAME
    (STDAIR_SAMPLE_DIR
    "/ond01.csv" );
00037
00038
00042 const std::string K_SIMCRS_DEFAULT_YIELD_INPUT_FILENAME
    (STDAIR_SAMPLE_DIR
    "/yieldstore01.csv" );
00043
00044
00048 const std::string K_SIMCRS_DEFAULT_FARE_INPUT_FILENAME
    (STDAIR_SAMPLE_DIR
    "/fare01.csv" );
00049
00050
```

```

00056 const bool K_SIMCRS_DEFAULT_BUILT_IN_INPUT =
00057     false;
00061 const std::string K_SIMCRS_DEFAULT_DB_USER ("dsim");
00062 const std::string K_SIMCRS_DEFAULT_DB_PASSWD ("dsim")
00063 ;
00063 const std::string K_SIMCRS_DEFAULT_DB_DBNAME ("
00064     sim_dsim");
00064 const std::string K_SIMCRS_DEFAULT_DB_HOST ("localhost"
00065 );
00065 const std::string K_SIMCRS_DEFAULT_DB_PORT ("3306");
00066
00067 // ////////// Parsing of Options & Configuration //////////
00068 // A helper function to simplify the main part.
00069 template<class T> std::ostream& operator<< (std::ostream& os,
00070     const std::vector<T>& v) {
00071     std::copy (v.begin(), v.end(), std::ostream_iterator<T> (std::cout, " "));
00072     return os;
00073 }
00074
00076 const int K_SIMCRS_EARLY_RETURN_STATUS = 99;
00077
00079 int readConfiguration (int argc, char* argv[],
00080     bool& ioIsBuiltin,
00081     stdair::Filename_T& ioScheduleInputFilename,
00082     stdair::Filename_T& ioOnDInputFilename,
00083     stdair::Filename_T& ioYieldInputFilename,
00084     stdair::Filename_T& ioFareInputFilename,
00085     stdair::Filename_T& ioLogFilename,
00086     std::string& ioDBUser, std::string& ioDBPasswd,
00087     std::string& ioDBHost, std::string& ioDBPort,
00088     std::string& ioDBDBName) {
00089     // Default for the built-in input
00090     ioIsBuiltin = K_SIMCRS_DEFAULT_BUILT_IN_INPUT;
00091
00092     // Declare a group of options that will be allowed only on command line
00093     boost::program_options::options_description generic ("Generic options");
00094     generic.add_options()
00095         ("prefix", "print installation prefix")
00096         ("version,v", "print version string")
00097         ("help,h", "produce help message");
00098
00099     // Declare a group of options that will be allowed both on command
00100     // line and in config file
00101     boost::program_options::options_description config ("Configuration");
00102     config.add_options()
00103         ("builtin,b",
00104             "The sample BOM tree can be either built-in or parsed from input files. In
00105             that latter case, the input files must be specified as well (e.g.,
00106             -s/--schedule, -o/--ond, -f/--fare, -y/--yield)")
00107         ("schedule,s",
00108             boost::program_options::value< std::string >(&ioScheduleInputFilename)->
00109             default_value(K_SIMCRS_DEFAULT_SCHEDULE_INPUT_FILENAME
00110             ),
00111             "(CVS) input file for the schedules")
00112         ("ond,o",
00113             boost::program_options::value< std::string >(&ioOnDInputFilename)->
00114             default_value(K_SIMCRS_DEFAULT_OND_INPUT_FILENAME),
00115             "(CVS) input file for the O&D definitions")
00116         ("yield,y",
00117             boost::program_options::value< std::string >(&ioYieldInputFilename)->
00118             default_value(K_SIMCRS_DEFAULT_YIELD_INPUT_FILENAME
00119             ),
00120             "(CVS) input file for the yields")
00121         ("fare,f",
00122             boost::program_options::value< std::string >(&ioFareInputFilename)->
00123             default_value(K_SIMCRS_DEFAULT_FARE_INPUT_FILENAME
00124             ),
00125             "(CVS) input file for the fares")
00126         ("log,l",
00127             boost::program_options::value< std::string >(&ioLogFilename)->
00128             default_value(K_SIMCRS_DEFAULT_LOG_FILENAME),
00129             "Filepath for the logs")
00130         ("user,u",
00131             boost::program_options::value< std::string >(&ioDBUser)->default_value(
00132             K_SIMCRS_DEFAULT_DB_USER),
00133             "SQL database username")
00134         ("passwd,p",
00135             boost::program_options::value< std::string >(&ioDBPasswd)->default_value(
00136             K_SIMCRS_DEFAULT_DB_PASSWD),
00137             "SQL database password")
00138         ("host,H",
00139             boost::program_options::value< std::string >(&ioDBHost)->default_value(
00140             K_SIMCRS_DEFAULT_DB_HOST),
00141             "SQL database hostname")
00142         ("port,P",
00143             boost::program_options::value< std::string >(&ioDBPort)->default_value(

```

```

    K_SIMCRS_DEFAULT_DB_PORT),
00131     "SQL database port")
00132     ("dbname,m",
00133     boost::program_options::value< std::string >(&ioDBDBName)->default_value(
    K_SIMCRS_DEFAULT_DB_DBNAME),
00134     "SQL database name")
00135     ;
00136
00137     // Hidden options, will be allowed both on command line and
00138     // in config file, but will not be shown to the user.
00139     boost::program_options::options_description hidden ("Hidden options");
00140     hidden.add_options()
00141         ("copyright",
00142         boost::program_options::value< std::vector<std::string> >(),
00143         "Show the copyright (license)");
00144
00145     boost::program_options::options_description cmdline_options;
00146     cmdline_options.add(generic).add(config).add(hidden);
00147
00148     boost::program_options::options_description config_file_options;
00149     config_file_options.add(config).add(hidden);
00150
00151     boost::program_options::options_description visible ("Allowed options");
00152     visible.add(generic).add(config);
00153
00154     boost::program_options::positional_options_description p;
00155     p.add ("copyright", -1);
00156
00157     boost::program_options::variables_map vm;
00158     boost::program_options::
00159         store (boost::program_options::command_line_parser (argc, argv).
00160             options (cmdline_options).positional(p).run(), vm);
00161
00162     std::ifstream ifs ("simcrs.cfg");
00163     boost::program_options::store (parse_config_file (ifs, config_file_options),
00164         vm);
00165     boost::program_options::notify (vm);
00166
00167     if (vm.count ("help")) {
00168         std::cout << visible << std::endl;
00169         return K_SIMCRS_EARLY_RETURN_STATUS;
00170     }
00171
00172     if (vm.count ("version")) {
00173         std::cout << PACKAGE_NAME << " , version " << PACKAGE_VERSION
00174         << std::endl;
00175         return K_SIMCRS_EARLY_RETURN_STATUS;
00176     }
00177
00178     if (vm.count ("prefix")) {
00179         std::cout << "Installation prefix: " << PREFIXDIR << std::endl;
00180         return K_SIMCRS_EARLY_RETURN_STATUS;
00181     }
00182
00183     if (vm.count ("builtin")) {
00184         ioIsBuiltin = true;
00185     }
00186     const std::string isBuiltinStr = (ioIsBuiltin == true)?"yes":"no";
00187     std::cout << "The BOM should be built-in? " << isBuiltinStr << std::endl;
00188
00189     //
00190     std::ostringstream oErrorMessageStr;
00191     oErrorMessageStr << "Either the -b/--builtin option, or the combination of "
00192         << "the -s/--schedule, -o/--ond, -f/--fare and -y/--yield "
00193         << "options must be specified";
00194
00195     if (ioIsBuiltin == false) {
00196         if (vm.count ("schedule")) {
00197             ioScheduleInputFilename = vm["schedule"].as< std::string >();
00198             std::cout << "Schedule input filename is: " << ioScheduleInputFilename
00199             << std::endl;
00200         } else {
00201             // The built-in option is not selected. However, no schedule input file
00202             // is specified
00203             std::cerr << oErrorMessageStr.str() << std::endl;
00204         }
00205
00206         if (vm.count ("ond")) {
00207             ioOnDInputFilename = vm["ond"].as< std::string >();
00208             std::cout << "O&D input filename is: " << ioOnDInputFilename << std::endl;
00209         } else {
00210             // The built-in option is not selected. However, no schedule input file
00211             // is specified
00212             std::cerr << oErrorMessageStr.str() << std::endl;
00213         }

```

```

00214     }
00215
00216     if (vm.count ("yield")) {
00217         ioYieldInputFilename = vm["yield"].as< std::string >();
00218         std::cout << "Yield input filename is: " << ioYieldInputFilename
00219             << std::endl;
00220
00221     } else {
00222         // The built-in option is not selected. However, no schedule input file
00223         // is specified
00224         std::cerr << oErrorMessageStr.str() << std::endl;
00225     }
00226
00227     if (vm.count ("fare")) {
00228         ioFareInputFilename = vm["fare"].as< std::string >();
00229         std::cout << "Fare input filename is: " << ioFareInputFilename
00230             << std::endl;
00231
00232     } else {
00233         // The built-in option is not selected. However, no schedule input file
00234         // is specified
00235         std::cerr << oErrorMessageStr.str() << std::endl;
00236     }
00237 }
00238
00239 if (vm.count ("log")) {
00240     ioLogFilename = vm["log"].as< std::string >();
00241     std::cout << "Log filename is: " << ioLogFilename << std::endl;
00242 }
00243
00244 if (vm.count ("user")) {
00245     ioDBUser = vm["user"].as< std::string >();
00246     std::cout << "SQL database user name is: " << ioDBUser << std::endl;
00247 }
00248
00249 if (vm.count ("passwd")) {
00250     ioDBPasswd = vm["passwd"].as< std::string >();
00251     //std::cout << "SQL database user password is: " << ioDBPasswd <<
std::endl;
00252 }
00253
00254 if (vm.count ("host")) {
00255     ioDBHost = vm["host"].as< std::string >();
00256     std::cout << "SQL database host name is: " << ioDBHost << std::endl;
00257 }
00258
00259 if (vm.count ("port")) {
00260     ioDBPort = vm["port"].as< std::string >();
00261     std::cout << "SQL database port number is: " << ioDBPort << std::endl;
00262 }
00263
00264 if (vm.count ("dbname")) {
00265     ioDBDBName = vm["dbname"].as< std::string >();
00266     std::cout << "SQL database name is: " << ioDBDBName << std::endl;
00267 }
00268
00269 return 0;
00270 }
00271
00272 // ////////// M A I N //////////
00273 int main (int argc, char* argv[]) {
00274
00275     // State whether the BOM tree should be built-in or parsed from an
00276     // input file
00277     bool isBuiltin;
00278
00279     // Schedule input filename
00280     stdair::Filename_T lScheduleInputFilename;
00281
00282     // O&D input filename
00283     stdair::Filename_T lOnDInputFilename;
00284
00285     // Yield input filename
00286     stdair::Filename_T lYieldInputFilename;
00287
00288     // Fare input filename
00289     stdair::Filename_T lFareInputFilename;
00290
00291     // Output log File
00292     stdair::Filename_T lLogFilename;
00293
00294     // SQL database parameters
00295     std::string lDBUser;
00296     std::string lDBPasswd;
00297     std::string lDBHost;
00298     std::string lDBPort;
00299     std::string lDBDBName;

```

```

00300
00301 // CRS code
00302 const SIMCRS::CRSCode_T lCRSCode ("1P");
00303
00304 // Call the command-line option parser
00305 const int lOptionParserStatus =
00306     readConfiguration (argc, argv, isBuiltin,
00307                         lScheduleInputFilename, lOnDInputFilename,
00308                         lYieldInputFilename, lFareInputFilename, lLogFilename,
00309                         lDBUser, lDBPasswd, lDBHost, lDBPort, lDBDBName);
00310
00311 if (lOptionParserStatus == K_SIMCRS_EARLY_RETURN_STATUS
00312 ) {
00313     return 0;
00314 }
00315 // Set the database parameters
00316 const stdair::BasDBParams lDBParams (lDBUser, lDBPasswd, lDBHost, lDBPort,
00317                                     lDBDBName);
00318
00319 // Set the log parameters
00320 std::ofstream logOutputFile;
00321 // Open and clean the log outputfile
00322 logOutputFile.open (lLogFilename.c_str());
00323 logOutputFile.clear();
00324
00325 // Initialise the list of classes/buckets
00326 const stdair::BasLogParams lLogParams (stdair::LOG::DEBUG, logOutputFile);
00327 SIMCRS::SIMCRS_Service simcrsService (lLogParams,
00328                                     lCRSCode);
00329
00330 // Check whether or not (CSV) input files should be read
00331 if (isBuiltin == true) {
00332     // Build the sample BOM tree
00333     simcrsService.buildSampleBom();
00334 } else {
00335     // Build the BOM tree from parsing input files
00336     const SIMFQT::FareFilePath lFareFilePath (lFareInputFilename);
00337     const AIRRAC::YieldFilePath lYieldFilePath (lYieldInputFilename);
00338     simcrsService.parseAndLoad (lScheduleInputFilename,
00339                                lOnDInputFilename,
00340                                lYieldFilePath, lFareFilePath);
00341 }
00342
00343 // TODO (issue #37707): instead of building a sample, read the parameters
00344 // from the command-line options, and build the corresponding
00345 // booking request
00346 const bool isForCRS = true;
00347 const stdair::BookingRequestStruct& lBookingRequest =
00348     simcrsService.buildSampleBookingRequest (isForCRS)
00349 ;
00350 // Calculate the travel solutions corresponding to the given booking request
00351 stdair::TravelSolutionList_T lTravelSolutionList =
00352     simcrsService.calculateSegmentPathList (
00353         lBookingRequest);
00354 // Check whether everything was fine
00355 if (lTravelSolutionList.empty() == true) {
00356     STDAIR_LOG_ERROR ("No travel solution has been found for: "
00357                       << lBookingRequest.display());
00358     return -1;
00359 }
00360 // Price the travel solution
00361 simcrsService.fareQuote (lBookingRequest, lTravelSolutionList);
00362
00363 // Choose a random travel solution: the first one.
00364 stdair::TravelSolutionStruct& lChosenTravelSolution =
00365     lTravelSolutionList.front();
00366
00367 // Get the segment path of the travel solution.
00368 const stdair::KeyList_T& lsegmentDateKeyList =
00369     lChosenTravelSolution.getSegmentPath();
00370
00371 const stdair::FareOptionList_T& lFareOptionList =
00372     lChosenTravelSolution.getFareOptionList();
00373
00374 // Check whether everything was fine
00375 if (lFareOptionList.empty() == true) {
00376     STDAIR_LOG_ERROR ("No fare option for the chosen travel solution: "
00377                       << lChosenTravelSolution.display());
00378     return -1;
00379 }
00380
00381

```



```

00382 //
00383 const stdair::FareOptionStruct& lFareOption = lFareOptionList.front();
00384 lChosenTravelSolution.setChosenFareOption (lFareOption);
00385
00386 // DEBUG
00387 const std::string& lSegmentDateKey = lsegmentDateKeyList.front();
00388 STDAIR_LOG_DEBUG ("The chosen travel solution is: " << lSegmentDateKey
00389                  << ", the fare is: " << lFareOption.getFare() << " Euros.")
;
00390
00391 // Make a booking (reminder: party size is 3)
00392 const stdair::PartySize_T lPartySize (3);
00393 const bool isSellSuccessful =
00394     simcrsService.sell (lChosenTravelSolution, lPartySize);
00395
00396 // DEBUG
00397 STDAIR_LOG_DEBUG ("Sale ('" << lBookingRequest << "'): "
00398                  << " successful? " << isSellSuccessful);
00399
00400 // DEBUG: Display the whole BOM tree
00401 const std::string& lCSVDump = simcrsService.csvDisplay();
00402 STDAIR_LOG_DEBUG (lCSVDump);
00403
00404 // Close the Log outputFile
00405 logOutputFile.close();
00406
00407 /*
00408     Note: as that program is not intended to be run on a server in
00409     production, it is better not to catch the exceptions. When it
00410     happens (that an exception is throwned), that way we get the
00411     call stack.
00412 */
00413
00414 return 0;
00415 }

```

## 23.23 simcrs/bom/BomAbstract.cpp File Reference

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

### Namespaces

- namespace [SIMCRS](#)

## 23.24 BomAbstract.cpp

```

00001 // //////////////////////////////////////
00002 // Import section
00003 // //////////////////////////////////////
00004 // SIMCRS
00005 #include <simcrs/bom/BomAbstract.hpp>
00006
00007 namespace SIMCRS {
00008
00009 }

```

## 23.25 simcrs/bom/BomAbstract.hpp File Reference

```
#include <iosfwd>
#include <string>
```

### Classes

- class [SIMCRS::BomAbstract](#)

## Namespaces

- namespace [SIMCRS](#)

## Functions

- template<class charT , class traits >  
std::basic\_ostream< charT,  
traits > &operator<< (std::basic\_ostream< charT, traits > &ioOut, const [SIMCRS::BomAbstract](#) &iBom)
- template<class charT , class traits >  
std::basic\_istream< charT,  
traits > &operator>> (std::basic\_istream< charT, traits > &ioIn, [SIMCRS::BomAbstract](#) &iBom)

## 23.25.1 Function Documentation

23.25.1.1 `template<class charT , class traits > std::basic_ostream<charT, traits>& operator<< ( std::basic_ostream< charT, traits > &ioOut, const SIMCRS::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](#).

23.25.1.2 `template<class charT , class traits > std::basic_istream<charT, traits>& operator>> ( std::basic_istream< charT, traits > &ioIn, SIMCRS::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 [SIMCRS::BomAbstract::fromStream\(\)](#).

## 23.26 BomAbstract.hpp

```
00001 #ifndef __SIMCRS_BOM_BOMABSTRACT_HPP
00002 #define __SIMCRS_BOM_BOMABSTRACT_HPP
00003
00004 // //////////////////////////////////////
00005 // Import section
00006 // //////////////////////////////////////
00007 // STL
00008 #include <iosfwd>
00009 #include <string>
00010
00011 namespace SIMCRS {
00012
00013     class BomAbstract {
00014     friend class FacBomAbstract;
00015     public:
00016         // ////////////////////////////////// Display support methods //////////////////////////////////
00017         virtual void toStream (std::ostream& ioOut) const = 0;
00021
00024         virtual void fromStream (std::istream& ioIn) = 0;
00025
00027         virtual std::string toString() const = 0;
00028
00031         virtual std::string describeKey() const = 0;
00032
00035         virtual std::string describeShortKey() const = 0;
00036
00037     protected:
00040         BomAbstract() {}
00041         BomAbstract(const BomAbstract&) {}
00042
00044         virtual ~BomAbstract() {}
00045     };
00046 }
00047
```

```

00053 template <class charT, class traits>
00054 inline
00055 std::basic_ostream<charT, traits>&
00056 operator<< (std::basic_ostream<charT, traits>& ioOut,
00057             const SIMCRS::BomAbstract& iBom) {
00063     std::basic_ostringstream<charT, traits> ostr;
00064     ostr.copyfmt (ioOut);
00065     ostr.width (0);
00066
00067     // Fill string stream
00068     iBom.toStream (ostr);
00069
00070     // Print string stream
00071     ioOut << ostr.str();
00072
00073     return ioOut;
00074 }
00075
00081 template <class charT, class traits>
00082 inline
00083 std::basic_istream<charT, traits>&
00084 operator>> (std::basic_istream<charT, traits>& ioIn,
00085             SIMCRS::BomAbstract& ioBom) {
00086     // Fill Bom object with input stream
00087     ioBom.fromStream (ioIn);
00088     return ioIn;
00089 }
00090
00091 #endif // __SIMCRS_BOM_BOMABSTRACT_HPP

```

## 23.27 simcrs/command/DistributionManager.cpp File Reference

```

#include <cassert>
#include <stdair/bom/FareOptionStruct.hpp>
#include <stdair/bom/TravelSolutionStruct.hpp>
#include <stdair/bom/CancellationStruct.hpp>
#include <stdair/service/Logger.hpp>
#include <airinv/AIRINV_Master_Service.hpp>
#include <simcrs/command/DistributionManager.hpp>

```

### Namespaces

- namespace [SIMCRS](#)

## 23.28 DistributionManager.cpp

```

00001 // //////////////////////////////////////
00002 // Import section
00003 // //////////////////////////////////////
00004 // STL
00005 #include <cassert>
00006 // StdAir
00007 #include <stdair/bom/FareOptionStruct.hpp>
00008 #include <stdair/bom/TravelSolutionStruct.hpp>
00009 #include <stdair/bom/CancellationStruct.hpp>
00010 #include <stdair/service/Logger.hpp>
00011 // Airline Inventory
00012 #include <airinv/AIRINV_Master_Service.hpp>
00013 // SimCRS
00014 #include <simcrs/command/DistributionManager.hpp>
00015 >
00016 namespace SIMCRS {
00017
00018     // //////////////////////////////////////
00019     void DistributionManager::
00020     calculateAvailability (AIRINV::AIRINV_Master_Service& ioAIRINV_Master_Service
00021
00022 ,
00023                          stdair::TravelSolutionList_T& ioTravelSolutionList,
00024                          const stdair::PartnershipTechnique&
00025                          iPartnershipTechnique) {
00023         for (stdair::TravelSolutionList_T::iterator itTS =
00024             ioTravelSolutionList.begin();

```

```

00025         itTS != ioTravelSolutionList.end(); ++itTS) {
00026     stdair::TravelSolutionStruct& lCurrentTravelSolution = *itTS;
00027
00028     // Forward the work to the dedicated service.
00029     ioAIRINV_Master_Service.calculateAvailability (lCurrentTravelSolution,
00030                                                 iPartnershipTechnique);
00031 }
00032 }
00033
00034 // //////////////////////////////////////
00035 bool DistributionManager::
00036 sell (AIRINV::AIRINV_Master_Service& ioAIRINV_Master_Service,
00037      const stdair::TravelSolutionStruct& iTravelSolution,
00038      const stdair::NbOfSeats_T& iPartySize) {
00039     bool hasSaleBeenSuccessful = false;
00040
00041     const stdair::KeyList_T& lSegmentDateKeyList =
00042         iTravelSolution.getSegmentPath();
00043     const stdair::FareOptionStruct& lChosenFareOption =
00044         iTravelSolution.getChosenFareOption ();
00045     const stdair::ClassList_StringList_T& lClassPath =
00046         lChosenFareOption.getClassPath();
00047     stdair::ClassList_StringList_T::const_iterator itClassKeyList =
00048         lClassPath.begin();
00049     for (stdair::KeyList_T::const_iterator itKey= lSegmentDateKeyList.begin();
00050          itKey != lSegmentDateKeyList.end(); ++itKey, ++itClassKeyList) {
00051         const std::string& lSegmentDateKey = *itKey;
00052
00053         // TODO: Removed this hardcoded.
00054         std::ostream ostr;
00055         const stdair::ClassList_String_T& lClassList = *itClassKeyList;
00056         assert (lClassList.size() > 0);
00057         ostr << lClassList.at(0);
00058         const stdair::ClassCode_T lClassCode (ostr.str());
00059
00060         hasSaleBeenSuccessful =
00061             ioAIRINV_Master_Service.sell (lSegmentDateKey, lClassCode, iPartySize);
00062     }
00063
00064     return hasSaleBeenSuccessful;
00065 }
00066
00067 // //////////////////////////////////////
00068 bool DistributionManager::
00069 playCancellation (AIRINV::AIRINV_Master_Service& ioAIRINV_Master_Service,
00070                  const stdair::CancellationStruct& iCancellation) {
00071     bool hasCancellationBeenSuccessful = false;
00072
00073     const stdair::PartySize_T& lPartySize = iCancellation.getPartySize();
00074     const stdair::KeyList_T& lSegmentDateKeyList =
00075         iCancellation.getSegmentPath();
00076     const stdair::ClassList_String_T& lClassList = iCancellation.getClassList();
00077
00078     stdair::ClassList_String_T::const_iterator itClass = lClassList.begin();
00079     for (stdair::KeyList_T::const_iterator itKey= lSegmentDateKeyList.begin();
00080          itKey != lSegmentDateKeyList.end(); ++itKey, ++itClass) {
00081         const std::string& lSegmentDateKey = *itKey;
00082
00083         // TODO: Removed this hardcoded.
00084         std::ostream ostr;
00085         ostr << *itClass;
00086         const stdair::ClassCode_T lClassCode (ostr.str());
00087
00088         hasCancellationBeenSuccessful =
00089             ioAIRINV_Master_Service.cancel (lSegmentDateKey, lClassCode,
00090                                           lPartySize);
00091     }
00092     return hasCancellationBeenSuccessful;
00093 }
00094 }

```

## 23.29 simcrs/command/DistributionManager.hpp File Reference

```

#include <stdair/stdair_basic_types.hpp>
#include <stdair/bom/TravelSolutionTypes.hpp>
#include <stdair/basic/PartnershipTechnique.hpp>
#include <airinv/AIRINV_Types.hpp>
#include <simcrs/SIMCRS_Types.hpp>

```

## Classes

- class [SIMCRS::DistributionManager](#)  
*Command wrapping the travel distribution (CRS/GDS) process.*

## Namespaces

- namespace [stdair](#)  
*Forward declarations.*
- namespace [AIRINV](#)
- namespace [SIMCRS](#)

## 23.30 DistributionManager.hpp

```

00001 #ifndef __SIMCRS_CMD_DISTRIBUTIONMANAGER_HPP
00002 #define __SIMCRS_CMD_DISTRIBUTIONMANAGER_HPP
00003
00004 // //////////////////////////////////////
00005 // Import section
00006 // //////////////////////////////////////
00007 // StdAir
00008 #include <stdair/stdair_basic_types.hpp>
00009 #include <stdair/bom/TravelSolutionTypes.hpp>
00010 #include <stdair/basic/PartnershipTechnique.hpp>
00011 // Airinv
00012 #include <airinv/AIRINV_Types.hpp>
00013 // Simcrs
00014 #include <simcrs/SIMCRS_Types.hpp>
00015
00016 // Forward declarations
00017 namespace stdair {
00018     struct TravelSolutionStruct;
00019     struct CancellationStruct;
00020 }
00021
00022 namespace AIRINV {
00023     class AIRINV_Master_Service;
00024 }
00025
00026 namespace SIMCRS {
00027
00031     class DistributionManager {
00032     friend class SIMCRS_Service;
00033     private:
00035         static void calculateAvailability (AIRINV::AIRINV_Master_Service&,
00036                                           stdair::TravelSolutionList_T&,
00037                                           const stdair::PartnershipTechnique&);
00038
00040         static bool sell (AIRINV::AIRINV_Master_Service&,
00041                          const stdair::TravelSolutionStruct&,
00042                          const stdair::NbOfSeats_T&);
00043
00045         static bool playCancellation (AIRINV::AIRINV_Master_Service&,
00046                                       const stdair::CancellationStruct&);
00047
00048     private:
00050         DistributionManager() {}
00051         DistributionManager(const DistributionManager
00052 &) {}
00053         ~DistributionManager() {}
00054     };
00055
00056 }
00057 #endif // __SIMCRS_CMD_DISTRIBUTIONMANAGER_HPP

```

## 23.31 simcrs/config/simcrs-paths.hpp File Reference

## Macros

- #define [PACKAGE](#) "simcrs"
- #define [PACKAGE\\_NAME](#) "SIMCRS"
- #define [PACKAGE\\_VERSION](#) "0.1.1"

- `#define PREFIXDIR "/usr"`
- `#define EXEC_PREFIX "/usr"`
- `#define BINDIR "/usr/bin"`
- `#define LIBDIR "/usr/lib"`
- `#define LIBEXECDIR "/usr/libexec"`
- `#define SBINDIR "/usr/sbin"`
- `#define SYSCONFDIR "/usr/etc"`
- `#define INCLUDEDIR "/usr/include"`
- `#define DATAROOTDIR "/usr/share"`
- `#define DATADIR "/usr/share"`
- `#define DOCDIR "/usr/share/doc/simcrs-0.1.1"`
- `#define MANDIR "/usr/share/man"`
- `#define INFODIR "/usr/share/info"`
- `#define HTMLDIR "/usr/share/doc/simcrs-0.1.1/html"`
- `#define PDFDIR "/usr/share/doc/simcrs-0.1.1/html"`
- `#define STDAIR_SAMPLE_DIR "/usr/share/stdair/samples"`

### 23.31.1 Macro Definition Documentation

#### 23.31.1.1 `#define PACKAGE "simcrs"`

Definition at line 4 of file [simcrs-paths.hpp](#).

#### 23.31.1.2 `#define PACKAGE_NAME "SIMCRS"`

Definition at line 5 of file [simcrs-paths.hpp](#).

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

#### 23.31.1.3 `#define PACKAGE_VERSION "0.1.1"`

Definition at line 6 of file [simcrs-paths.hpp](#).

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

#### 23.31.1.4 `#define PREFIXDIR "/usr"`

Definition at line 7 of file [simcrs-paths.hpp](#).

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

#### 23.31.1.5 `#define EXEC_PREFIX "/usr"`

Definition at line 8 of file [simcrs-paths.hpp](#).

#### 23.31.1.6 `#define BINDIR "/usr/bin"`

Definition at line 9 of file [simcrs-paths.hpp](#).

#### 23.31.1.7 `#define LIBDIR "/usr/lib"`

Definition at line 10 of file [simcrs-paths.hpp](#).

#### 23.31.1.8 `#define LIBEXECDIR "/usr/libexec"`

Definition at line 11 of file [simcrs-paths.hpp](#).

#### 23.31.1.9 `#define SBINDIR "/usr/sbin"`

Definition at line 12 of file [simcrs-paths.hpp](#).

23.31.1.10 `#define SYSCONFDIR "/usr/etc"`

Definition at line 13 of file [simcrs-paths.hpp](#).

23.31.1.11 `#define INCLUDEDIR "/usr/include"`

Definition at line 14 of file [simcrs-paths.hpp](#).

23.31.1.12 `#define DATAROOTDIR "/usr/share"`

Definition at line 15 of file [simcrs-paths.hpp](#).

23.31.1.13 `#define DATADIR "/usr/share"`

Definition at line 16 of file [simcrs-paths.hpp](#).

23.31.1.14 `#define DOCDIR "/usr/share/doc/simcrs-0.1.1"`

Definition at line 17 of file [simcrs-paths.hpp](#).

23.31.1.15 `#define MANDIR "/usr/share/man"`

Definition at line 18 of file [simcrs-paths.hpp](#).

23.31.1.16 `#define INFODIR "/usr/share/info"`

Definition at line 19 of file [simcrs-paths.hpp](#).

23.31.1.17 `#define HTMLDIR "/usr/share/doc/simcrs-0.1.1/html"`

Definition at line 20 of file [simcrs-paths.hpp](#).

23.31.1.18 `#define PDFDIR "/usr/share/doc/simcrs-0.1.1/html"`

Definition at line 21 of file [simcrs-paths.hpp](#).

23.31.1.19 `#define STDAIR_SAMPLE_DIR "/usr/share/stdair/samples"`

Definition at line 22 of file [simcrs-paths.hpp](#).

## 23.32 simcrs-paths.hpp

```
00001 #ifndef __SIMCRS_PATHS_HPP__
00002 #define __SIMCRS_PATHS_HPP__
00003
00004 #define PACKAGE "simcrs"
00005 #define PACKAGE_NAME "SIMCRS"
00006 #define PACKAGE_VERSION "0.1.1"
00007 #define PREFIXDIR "/usr"
00008 #define EXEC_PREFIX "/usr"
00009 #define BINDIR "/usr/bin"
00010 #define LIBDIR "/usr/lib"
00011 #define LIBEXECDIR "/usr/libexec"
00012 #define SBINDIR "/usr/sbin"
00013 #define SYSCONFDIR "/usr/etc"
00014 #define INCLUDEDIR "/usr/include"
00015 #define DATAROOTDIR "/usr/share"
00016 #define DATADIR "/usr/share"
00017 #define DOCDIR "/usr/share/doc/simcrs-0.1.1"
00018 #define MANDIR "/usr/share/man"
00019 #define INFODIR "/usr/share/info"
00020 #define HTMLDIR "/usr/share/doc/simcrs-0.1.1/html"
00021 #define PDFDIR "/usr/share/doc/simcrs-0.1.1/html"
00022 #define STDAIR_SAMPLE_DIR "/usr/share/stdair/samples"
00023
00024 #endif // __SIMCRS_PATHS_HPP__
```

## 23.33 simcrs/config/simcrs-paths.hpp.in File Reference

### Macros

- `#define __SIMCRS_PATHS_HPP__`
- `#define PACKAGE "@PACKAGE@"`
- `#define PACKAGE_NAME "@PACKAGE_NAME@"`
- `#define PACKAGE_VERSION "@PACKAGE_VERSION@"`
- `#define PREFIXDIR "@prefix@"`
- `#define EXEC_PREFIX "@exec_prefix@"`
- `#define BINDIR "@bindir@"`
- `#define LIBDIR "@libdir@"`
- `#define LIBEXECDIR "@libexecdir@"`
- `#define SBINDIR "@sbindir@"`
- `#define SYSCONFDIR "@sysconfdir@"`
- `#define INCLUDEDIR "@includedir@"`
- `#define DATAROOTDIR "@datarootdir@"`
- `#define DATADIR "@datadir@"`
- `#define DOCDIR "@docdir@"`
- `#define MANDIR "@mandir@"`
- `#define INFODIR "@infodir@"`
- `#define HTMLDIR "@htmldir@"`
- `#define PDFDIR "@pdfdir@"`
- `#define STDAIR_SAMPLE_DIR "@sampledir@"`

#### 23.33.1 Macro Definition Documentation

##### 23.33.1.1 `#define __SIMCRS_PATHS_HPP__`

Definition at line 2 of file [simcrs-paths.hpp.in](#).

##### 23.33.1.2 `#define PACKAGE "@PACKAGE@"`

Definition at line 4 of file [simcrs-paths.hpp.in](#).

##### 23.33.1.3 `#define PACKAGE_NAME "@PACKAGE_NAME@"`

Definition at line 5 of file [simcrs-paths.hpp.in](#).

##### 23.33.1.4 `#define PACKAGE_VERSION "@PACKAGE_VERSION@"`

Definition at line 6 of file [simcrs-paths.hpp.in](#).

##### 23.33.1.5 `#define PREFIXDIR "@prefix@"`

Definition at line 7 of file [simcrs-paths.hpp.in](#).

##### 23.33.1.6 `#define EXEC_PREFIX "@exec_prefix@"`

Definition at line 8 of file [simcrs-paths.hpp.in](#).

##### 23.33.1.7 `#define BINDIR "@bindir@"`

Definition at line 9 of file [simcrs-paths.hpp.in](#).

##### 23.33.1.8 `#define LIBDIR "@libdir@"`

Definition at line 10 of file [simcrs-paths.hpp.in](#).



23.33.1.9 `#define LIBEXECDIR "@libexecdir@"`

Definition at line 11 of file [simcrs-paths.hpp.in](#).

23.33.1.10 `#define SBINDIR "@sbindir@"`

Definition at line 12 of file [simcrs-paths.hpp.in](#).

23.33.1.11 `#define SYSCONFDIR "@sysconfdir@"`

Definition at line 13 of file [simcrs-paths.hpp.in](#).

23.33.1.12 `#define INCLUDEDIR "@includedir@"`

Definition at line 14 of file [simcrs-paths.hpp.in](#).

23.33.1.13 `#define DATAROOTDIR "@datarootdir@"`

Definition at line 15 of file [simcrs-paths.hpp.in](#).

23.33.1.14 `#define DATADIR "@datadir@"`

Definition at line 16 of file [simcrs-paths.hpp.in](#).

23.33.1.15 `#define DOCDIR "@docdir@"`

Definition at line 17 of file [simcrs-paths.hpp.in](#).

23.33.1.16 `#define MANDIR "@mandir@"`

Definition at line 18 of file [simcrs-paths.hpp.in](#).

23.33.1.17 `#define INFODIR "@infodir@"`

Definition at line 19 of file [simcrs-paths.hpp.in](#).

23.33.1.18 `#define HTMLDIR "@htmldir@"`

Definition at line 20 of file [simcrs-paths.hpp.in](#).

23.33.1.19 `#define PDFDIR "@pdfdir@"`

Definition at line 21 of file [simcrs-paths.hpp.in](#).

23.33.1.20 `#define STDAIR_SAMPLE_DIR "@sampledir@"`

Definition at line 22 of file [simcrs-paths.hpp.in](#).

## 23.34 simcrs-paths.hpp.in

```
00001 #ifndef __SIMCRS_PATHS_HPP__
00002 #define __SIMCRS_PATHS_HPP__
00003
00004 #define PACKAGE "@PACKAGE@"
00005 #define PACKAGE_NAME "@PACKAGE_NAME@"
00006 #define PACKAGE_VERSION "@PACKAGE_VERSION@"
00007 #define PREFIXDIR "@prefix@"
00008 #define EXEC_PREFIX "@exec_prefix@"
00009 #define BINDIR "@bindir@"
00010 #define LIBDIR "@libdir@"
00011 #define LIBEXECDIR "@libexecdir@"
00012 #define SBINDIR "@sbindir@"
00013 #define SYSCONFDIR "@sysconfdir@"
00014 #define INCLUDEDIR "@includedir@"
00015 #define DATAROOTDIR "@datarootdir@"
00016 #define DATADIR "@datadir@"
```

```

00017 #define DOCDIR "@docdir@"
00018 #define MANDIR "@mandir@"
00019 #define INFODIR "@infodir@"
00020 #define HTMLDIR "@htmldir@"
00021 #define PDFDIR "@pdfdir@"
00022 #define STDAIR_SAMPLE_DIR "@sampledir@"
00023
00024 #endif // __SIMCRS_PATHS_HPP__

```

## 23.35 simcrs/factory/FacBomAbstract.cpp File Reference

```

#include <cassert>
#include <sstream>
#include <boost/functional/hash/hash.hpp>
#include <simcrs/bom/BomAbstract.hpp>
#include <simcrs/factory/FacBomAbstract.hpp>

```

### Namespaces

- namespace [SIMCRS](#)

## 23.36 FacBomAbstract.cpp

```

00001 // //////////////////////////////////////
00002 // Import section
00003 // //////////////////////////////////////
00004 // STL
00005 #include <cassert>
00006 #include <sstream>
00007 // Boost (STL Extension)
00008 #include <boost/functional/hash/hash.hpp>
00009 // Simcrs
00010 #include <simcrs/bom/BomAbstract.hpp>
00011 #include <simcrs/factory/FacBomAbstract.hpp>
00012
00013 namespace SIMCRS {
00014
00015     // //////////////////////////////////////
00016     FacBomAbstract::~FacBomAbstract() {
00017         clean ();
00018     }
00019
00020     // //////////////////////////////////////
00021     void FacBomAbstract::clean() {
00022         for (BomPool_T::iterator itBom = _pool.begin();
00023              itBom != _pool.end(); itBom++) {
00024             BomAbstract* currentBom_ptr = *itBom;
00025             assert (currentBom_ptr != NULL);
00026
00027             delete (currentBom_ptr); currentBom_ptr = NULL;
00028         }
00029
00030         // Empty the pool of Factories
00031         _pool.clear();
00032     }
00033
00034     // //////////////////////////////////////
00035     std::size_t FacBomAbstract::getID (const BomAbstract
00036 * iBomAbstract_ptr) {
00037         const void* lPtr = iBomAbstract_ptr;
00038         boost::hash<const void*> ptr_hash;
00039         const std::size_t lID = ptr_hash (lPtr);
00040         return lID;
00041     }
00042
00043     // //////////////////////////////////////
00044     std::size_t FacBomAbstract::getID (const BomAbstract
00045 & iBomAbstract) {
00046         return getID (&iBomAbstract);
00047     }
00048
00049     // //////////////////////////////////////
00050     std::string FacBomAbstract::getIDString(const
00051 BomAbstract* iBomAbstract_ptr) {

```

```

00049     const std::size_t lID = getID (iBomAbstract_ptr);
00050     std::ostringstream ostr;
00051     ostr << lID;
00052     return ostr.str();
00053 }
00054
00055 // //////////////////////////////////////
00056 std::string FacBomAbstract::getIDString (const
BomAbstract& iBomAbstract) {
00057     return getIDString (&iBomAbstract);
00058 }
00059
00060 }

```

## 23.37 simcrs/factory/FacBomAbstract.hpp File Reference

```

#include <string>
#include <vector>

```

### Classes

- class [SIMCRS::FacBomAbstract](#)

### Namespaces

- namespace [SIMCRS](#)

## 23.38 FacBomAbstract.hpp

```

00001 #ifndef __SIMCRS_FAC_FACBOMABSTRACT_HPP
00002 #define __SIMCRS_FAC_FACBOMABSTRACT_HPP
00003
00004 // //////////////////////////////////////
00005 // Import section
00006 // //////////////////////////////////////
00007 // STL
00008 #include <string>
00009 #include <vector>
00010
00011 namespace SIMCRS {
00012
00013     // Forward declarations
00014     class BomAbstract;
00015
00016     class FacBomAbstract {
00017     friend class FacSupervisor;
00018     public:
00019
00020         typedef std::vector<BomAbstract*> BomPool_T;
00021
00022         static std::size_t getID (const BomAbstract*);
00023
00024         static std::size_t getID (const BomAbstract&);
00025
00026         static std::string getIDString (const BomAbstract*);
00027
00028         static std::string getIDString (const BomAbstract&);
00029
00030     protected:
00031         FacBomAbstract() {}
00032         FacBomAbstract(const FacBomAbstract&) {}
00033
00034         virtual ~FacBomAbstract();
00035
00036     private:
00037         void clean();
00038
00039     protected:
00040         BomPool_T _pool;
00041     };
00042 }
00043 #endif // __SIMCRS_FAC_FACBOMABSTRACT_HPP

```

## 23.39 simcrs/factory/FacServiceAbstract.cpp File Reference

```
#include <cassert>
#include <simcrs/service/ServiceAbstract.hpp>
#include <simcrs/factory/FacServiceAbstract.hpp>
```

### Namespaces

- namespace [SIMCRS](#)

## 23.40 FacServiceAbstract.cpp

```
00001 // //////////////////////////////////////
00002 // Import section
00003 // //////////////////////////////////////
00004 // STL
00005 #include <cassert>
00006 // SIMCRS
00007 #include <simcrs/service/ServiceAbstract.hpp>
00008 #include <simcrs/factory/FacServiceAbstract.hpp>
00009 >
00010 namespace SIMCRS {
00011
00012 // //////////////////////////////////////
00013 FacServiceAbstract::~FacServiceAbstract
00014 () {
00015     clean ();
00016 }
00017
00018 // //////////////////////////////////////
00019 void FacServiceAbstract::clean() {
00020     for (ServicePool_T::iterator itService = _pool.begin();
00021          itService != _pool.end(); itService++) {
00022         ServiceAbstract* currentService_ptr = *itService;
00023         assert (currentService_ptr != NULL);
00024         delete (currentService_ptr); currentService_ptr = NULL;
00025     }
00026
00027     // Empty the pool of Service Factories
00028     _pool.clear();
00029 }
00030
00031 }
```

## 23.41 simcrs/factory/FacServiceAbstract.hpp File Reference

```
#include <vector>
```

### Classes

- class [SIMCRS::FacServiceAbstract](#)

### Namespaces

- namespace [SIMCRS](#)

## 23.42 FacServiceAbstract.hpp

```
00001 #ifndef __SIMCRS_FAC_FACSERVICEABSTRACT_HPP
00002 #define __SIMCRS_FAC_FACSERVICEABSTRACT_HPP
00003
00004 // //////////////////////////////////////
```

```

00005 // Import section
00006 // //////////////////////////////////////
00007 // STL
00008 #include <vector>
00009
00010 namespace SIMCRS {
00011
00012     // Forward declarations
00013     class ServiceAbstract;
00014
00016     class FacServiceAbstract {
00017     public:
00018
00020         typedef std::vector<ServiceAbstract*> ServicePool_T;
00021
00023         virtual ~FacServiceAbstract();
00024
00026         void clean();
00027
00028     protected:
00031         FacServiceAbstract() {}
00032
00034         ServicePool_T _pool;
00035     };
00036
00037 }
00038 #endif // __SIMCRS_FAC_FACSERVICEABSTRACT_HPP

```

## 23.43 simcrs/factory/FacSimcrsServiceContext.cpp File Reference

```

#include <cassert>
#include <simcrs/factory/FacSupervisor.hpp>
#include <simcrs/factory/FacSimcrsServiceContext.hpp>
#include <simcrs/service/SIMCRS_ServiceContext.hpp>

```

### Namespaces

- namespace [SIMCRS](#)

## 23.44 FacSimcrsServiceContext.cpp

```

00001 // //////////////////////////////////////
00002 // Import section
00003 // //////////////////////////////////////
00004 // STL
00005 #include <cassert>
00006 // SIMCRS Common
00007 #include <simcrs/factory/FacSupervisor.hpp>
00008 #include <simcrs/factory/FacSimcrsServiceContext.hpp>
00009 #include <simcrs/service/SIMCRS_ServiceContext.hpp>
00010
00011 namespace SIMCRS {
00012
00013     FacSimcrsServiceContext* FacSimcrsServiceContext::_instance = NULL;
00014
00015     // //////////////////////////////////////
00016     FacSimcrsServiceContext::~FacSimcrsServiceContext
00017     () {
00018         _instance = NULL;
00019     }
00020
00021     // //////////////////////////////////////
00022     FacSimcrsServiceContext&
00023     FacSimcrsServiceContext::instance () {
00024
00025         if (_instance == NULL) {
00026             _instance = new FacSimcrsServiceContext();
00027             assert (_instance != NULL);
00028
00029             FacSupervisor::instance().registerServiceFactory (
00030                 _instance);
00031         }
00032         return *_instance;
00033     }
00034 }

```

```

00030     }
00031
00032     // //////////////////////////////////////
00033     SIMCRS_ServiceContext& FacSimcrsServiceContext::
00034     create (const std::string& iTravelDatabaseName) {
00035         SIMCRS_ServiceContext* aSIMCRS_ServiceContext_ptr =
00036         NULL;
00037         aSIMCRS_ServiceContext_ptr =
00038         new SIMCRS_ServiceContext (iTravelDatabaseName);
00039         assert (aSIMCRS_ServiceContext_ptr != NULL);
00040
00041         // The new object is added to the Bom pool
00042         _pool.push_back (aSIMCRS_ServiceContext_ptr);
00043
00044         return *aSIMCRS_ServiceContext_ptr;
00045     }
00046
00047 }

```

## 23.45 simcrs/factory/FacSimcrsServiceContext.hpp File Reference

```

#include <string>
#include <simcrs/factory/FacServiceAbstract.hpp>

```

### Classes

- class [SIMCRS::FacSimcrsServiceContext](#)

### Namespaces

- namespace [SIMCRS](#)

## 23.46 FacSimcrsServiceContext.hpp

```

00001 #ifndef __SIMCRS_FAC_FACSIMCRSSERVICECONTEXT_HPP
00002 #define __SIMCRS_FAC_FACSIMCRSSERVICECONTEXT_HPP
00003
00004 // //////////////////////////////////////
00005 // Import section
00006 // //////////////////////////////////////
00007 // STL
00008 #include <string>
00009 // Simcrs
00010 #include <simcrs/factory/FacServiceAbstract.hpp>
00011
00012 namespace SIMCRS {
00013
00014     class SIMCRS_ServiceContext;
00015
00016     class FacSimcrsServiceContext : public
00017     FacServiceAbstract {
00018     public:
00019
00020         static FacSimcrsServiceContext& instance();
00021
00022         ~FacSimcrsServiceContext();
00023
00024         SIMCRS_ServiceContext& create (const std::string
00025         & iTravelDatabaseName);
00026
00027     protected:
00028         FacSimcrsServiceContext () {}
00029
00030     private:
00031         static FacSimcrsServiceContext* _instance;
00032     };
00033
00034 }
00035 #endif // __SIMCRS_FAC_FACSIMCRSSERVICECONTEXT_HPP

```

## 23.47 simcrs/factory/FacSupervisor.cpp File Reference

```
#include <cassert>
#include <simcrs/factory/FacBomAbstract.hpp>
#include <simcrs/factory/FacServiceAbstract.hpp>
#include <simcrs/factory/FacSupervisor.hpp>
```

## Namespaces

- namespace [SIMCRS](#)

## 23.48 FacSupervisor.cpp

```
00001 // //////////////////////////////////////
00002 // Import section
00003 // //////////////////////////////////////
00004 // STL
00005 #include <cassert>
00006 // SIMCRS
00007 #include <simcrs/factory/FacBomAbstract.hpp>
00008 #include <simcrs/factory/FacServiceAbstract.hpp>
00009 #include <simcrs/factory/FacSupervisor.hpp>
00010
00011 namespace SIMCRS {
00012
00013     FacSupervisor* FacSupervisor::_instance = NULL;
00014
00015     // //////////////////////////////////////
00016     FacSupervisor::FacSupervisor () {
00017     }
00018
00019     // //////////////////////////////////////
00020     FacSupervisor& FacSupervisor::instance()
00021     {
00022         if (_instance == NULL) {
00023             _instance = new FacSupervisor();
00024         }
00025         return *_instance;
00026     }
00027
00028     // //////////////////////////////////////
00029     void FacSupervisor::
00030     registerBomFactory (FacBomAbstract*
00031     ioFacBomAbstract_ptr) {
00032         _bomPool.push_back (ioFacBomAbstract_ptr);
00033     }
00034
00035     // //////////////////////////////////////
00036     void FacSupervisor::
00037     registerServiceFactory (FacServiceAbstract
00038     * ioFacServiceAbstract_ptr) {
00039         _svcPool.push_back (ioFacServiceAbstract_ptr);
00040     }
00041
00042     // //////////////////////////////////////
00043     FacSupervisor::~FacSupervisor() {
00044         cleanBomLayer();
00045         cleanServiceLayer();
00046     }
00047
00048     // //////////////////////////////////////
00049     void FacSupervisor::cleanBomLayer() {
00050         for (BomFactoryPool_T::const_iterator itFactory = _bomPool.begin();
00051             itFactory != _bomPool.end(); itFactory++) {
00052             const FacBomAbstract* currentFactory_ptr = *itFactory;
00053             assert (currentFactory_ptr != NULL);
00054             delete (currentFactory_ptr); currentFactory_ptr = NULL;
00055         }
00056         // Empty the pool of Bom Factories
00057         _bomPool.clear();
00058     }
00059
00060     // //////////////////////////////////////
```

```

00061 void FacSupervisor::cleanServiceLayer() {
00062     for (ServiceFactoryPool_T::const_iterator itFactory = _svcPool.begin();
00063          itFactory != _svcPool.end(); itFactory++) {
00064         const FacServiceAbstract* currentFactory_ptr = *
itFactory;
00065         assert (currentFactory_ptr != NULL);
00066         delete (currentFactory_ptr); currentFactory_ptr = NULL;
00067     }
00068 }
00069 // Empty the pool of Service Factories
00070 _svcPool.clear();
00071 }
00072 }
00073 // //////////////////////////////////////
00074 void FacSupervisor::cleanFactory () {
00075     if (_instance != NULL) {
00076         _instance->cleanBomLayer();
00077         _instance->cleanServiceLayer();
00078     }
00079     delete (_instance); _instance = NULL;
00080 }
00081 }
00082 }
00083 }

```

## 23.49 simcrs/factory/FacSupervisor.hpp File Reference

```
#include <vector>
```

### Classes

- class [SIMCRS::FacSupervisor](#)

### Namespaces

- namespace [SIMCRS](#)

## 23.50 FacSupervisor.hpp

```

00001 #ifndef __SIMCRS_FAC_FACSUPERVISOR_HPP
00002 #define __SIMCRS_FAC_FACSUPERVISOR_HPP
00003
00004 // //////////////////////////////////////
00005 // Import section
00006 // //////////////////////////////////////
00007 // STL
00008 #include <vector>
00009
00010 namespace SIMCRS {
00011
00012     // Forward declarations
00013     class FacBomAbstract;
00014     class FacServiceAbstract;
00015
00016     class FacSupervisor {
00017     public:
00018
00019         typedef std::vector<FacBomAbstract*> BomFactoryPool_T;
00020         typedef std::vector<FacServiceAbstract*> ServiceFactoryPool_T;
00021
00022     };
00023
00024     static FacSupervisor& instance();
00025
00026     void registerBomFactory (FacBomAbstract*);
00027
00028     void registerServiceFactory (FacServiceAbstract
00029 *) ;
00030
00031     void cleanBomLayer();
00032
00033     void cleanServiceLayer();
00034
00035     static void cleanFactory ();

```



```

00054
00058     ~FacSupervisor();
00059
00060
00061 protected:
00062     FacSupervisor ();
00066     FacSupervisor (const FacSupervisor&) {}
00067
00068
00069 private:
00071     static FacSupervisor* _instance;
00072
00074     BomFactoryPool_T _bomPool;
00075
00077     ServiceFactoryPool_T _svcPool;
00078 };
00079 }
00080 #endif // __SIMCRS_FAC_FAC_SUPERVISOR_HPP

```

## 23.51 simcrs/service/ServiceAbstract.cpp File Reference

```
#include <simcrs/service/ServiceAbstract.hpp>
```

### Namespaces

- namespace [SIMCRS](#)

## 23.52 ServiceAbstract.cpp

```

00001 // //////////////////////////////////////
00002 // Import section
00003 // //////////////////////////////////////
00004 // SIMCRS
00005 #include <simcrs/service/ServiceAbstract.hpp>
00006
00007 namespace SIMCRS {
00008
00009 }

```

## 23.53 simcrs/service/ServiceAbstract.hpp File Reference

```
#include <iosfwd>
```

### Classes

- class [SIMCRS::ServiceAbstract](#)

### Namespaces

- namespace [SIMCRS](#)

### Functions

- `template<class charT, class traits>`  
`std::basic_ostream< charT,`  
`traits > & operator<< (std::basic_ostream< charT, traits > &ioOut, const SIMCRS::ServiceAbstract &i-`  
`Service)`
- `template<class charT, class traits>`  
`std::basic_istream< charT,`  
`traits > & operator>> (std::basic_istream< charT, traits > &ioIn, SIMCRS::ServiceAbstract &ioService)`

## 23.53.1 Function Documentation

**23.53.1.1** `template<class charT , class traits > std::basic_ostream<charT, traits>& operator<< ( std::basic_ostream< charT, traits > & ioOut, const SIMCRS::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](#).

**23.53.1.2** `template<class charT , class traits > std::basic_istream<charT, traits>& operator>> ( std::basic_istream< charT, traits > & ioIn, SIMCRS::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 [SIMCRS::ServiceAbstract::fromStream\(\)](#).

## 23.54 ServiceAbstract.hpp

```

00001 #ifndef __SIMCRS_SVC_SERVICEABSTRACT_HPP
00002 #define __SIMCRS_SVC_SERVICEABSTRACT_HPP
00003
00004 // //////////////////////////////////////
00005 // Import section
00006 // //////////////////////////////////////
00007 // STL
00008 #include <iosfwd>
00009 // #include <sstream>
00010
00011 namespace SIMCRS {
00012
00013     class ServiceAbstract {
00014     public:
00015
00016         virtual ~ServiceAbstract() {}
00017
00018         virtual void toStream (std::ostream& ioOut) const {}
00019
00020         virtual void fromStream (std::istream& ioIn) {}
00021
00022     protected:
00023         ServiceAbstract() {}
00024     };
00025
00026 template <class charT, class traits>
00027 inline
00028 std::basic_ostream<charT, traits>&
00029 operator<< (std::basic_ostream<charT, traits>& ioOut,
00030            const SIMCRS::ServiceAbstract& iService) {
00031     std::basic_ostringstream<charT, traits> ostr;
00032     ostr.copyfmt (ioOut);
00033     ostr.width (0);
00034     // Fill string stream
00035     iService.toStream (ostr);
00036     // Print string stream
00037     ioOut << ostr.str();
00038     return ioOut;
00039 }
00040
00041 template <class charT, class traits>
00042 inline
00043 std::basic_istream<charT, traits>&
00044 operator>> (std::basic_istream<charT, traits>& ioIn,
00045            SIMCRS::ServiceAbstract& ioService) {
00046     // Fill Service object with input stream
00047     ioService.fromStream (ioIn);
00048     return ioIn;
00049 }
00050
00051 #endif // __SIMCRS_SVC_SERVICEABSTRACT_HPP

```

**23.55 simcrs/service/SIMCRS\_Service.cpp File Reference**

```

#include <cassert>
#include <sstream>
#include <boost/make_shared.hpp>
#include <stdair/stdair_exceptions.hpp>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/basic/BasChronometer.hpp>
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/bom/BomManager.hpp>
#include <stdair/bom/BookingRequestStruct.hpp>
#include <stdair/bom/TravelSolutionStruct.hpp>
#include <stdair/bom/CancellationStruct.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/bom/Inventory.hpp>
#include <stdair/service/Logger.hpp>
#include <stdair/STDAIR_Service.hpp>
#include <airinv/AIRINV_Master_Service.hpp>
#include <airsched/AIRSCHEM_Service.hpp>
#include <simfqt/SIMFQT_Service.hpp>
#include <simcrs/basic/BasConst_SIMCRS_Service.hpp>
#include <simcrs/command/DistributionManager.hpp>
#include <simcrs/factory/FacSimcrsServiceContext.hpp>
#include <simcrs/service/SIMCRS_ServiceContext.hpp>
#include <simcrs/SIMCRS_Service.hpp>

```

**Namespaces**

- namespace [SIMCRS](#)

**23.56 SIMCRS\_Service.cpp**

```

00001 // //////////////////////////////////////
00002 // Import section
00003 // //////////////////////////////////////
00004 // STL
00005 #include <cassert>
00006 #include <sstream>
00007 // Boost
00008 #include <boost/make_shared.hpp>
00009 // Standard Airline Object Model
00010 #include <stdair/stdair_exceptions.hpp>
00011 #include <stdair/stdair_basic_types.hpp>
00012 #include <stdair/basic/BasChronometer.hpp>
00013 #include <stdair/basic/BasFileMgr.hpp>
00014 #include <stdair/bom/BomManager.hpp>
00015 #include <stdair/bom/BookingRequestStruct.hpp>
00016 #include <stdair/bom/TravelSolutionStruct.hpp>
00017 #include <stdair/bom/CancellationStruct.hpp>
00018 #include <stdair/bom/BomRoot.hpp>
00019 #include <stdair/bom/Inventory.hpp>
00020 #include <stdair/service/Logger.hpp>
00021 #include <stdair/STDAIR_Service.hpp>
00022 // Airline Inventory
00023 #include <airinv/AIRINV_Master_Service.hpp>
00024 // Airline Schedule
00025 #include <airsched/AIRSCHEM_Service.hpp>
00026 // Fare Quote
00027 #include <simfqt/SIMFQT_Service.hpp>
00028 // SimCRS
00029 #include <simcrs/basic/BasConst_SIMCRS_Service.hpp>
00030 >
00031 #include <simcrs/command/DistributionManager.hpp>
00032 >
00033 #include <simcrs/factory/FacSimcrsServiceContext.hpp>
00034 >
00035 #include <simcrs/service/SIMCRS_ServiceContext.hpp>
00036 >

```

```

00033 #include <simcrs/SIMCRS_Service.hpp>
00034
00035 namespace SIMCRS {
00036
00037     // //////////////////////////////////////
00038     SIMCRS_Service::SIMCRS_Service() : _simcrsServiceContext (NULL) {
00039         assert (false);
00040     }
00041
00042     // //////////////////////////////////////
00043     SIMCRS_Service::SIMCRS_Service (const SIMCRS_Service& iService) {
00044         assert (false);
00045     }
00046
00047     // //////////////////////////////////////
00048     SIMCRS_Service::SIMCRS_Service (const stdair::BasLogParams& iLogParams,
00049                                     const CRSCode_T& iCRSCode)
00050         : _simcrsServiceContext (NULL) {
00051
00052         // Initialise the StdAir service handler
00053         stdair::STDAIR_ServicePtr_T lSTDAIR_Service_ptr =
00054             initStdAirService (iLogParams);
00055
00056         // Initialise the service context
00057         initServiceContext (iCRSCode);
00058
00059         // Add the StdAir service context to the SimCRS service context
00060         // \note SIMCRS owns the STDAIR service resources here.
00061         const bool ownStdairService = true;
00062         addStdAirService (lSTDAIR_Service_ptr, ownStdairService);
00063
00064         // Initialise the SimFQT service.
00065         initSIMFQTService();
00066
00067         // Initialise the AirSched service.
00068         initAIRSCHEDService();
00069
00070         // Initialise the AirInv service.
00071         initAIRINVService();
00072
00073         // Initialise the (remaining of the) context
00074         initSimcrsService();
00075     }
00076
00077     // //////////////////////////////////////
00078     SIMCRS_Service::SIMCRS_Service (const stdair::BasLogParams& iLogParams,
00079                                     const stdair::BasDBParams& iDBParams,
00080                                     const CRSCode_T& iCRSCode)
00081         : _simcrsServiceContext (NULL) {
00082
00083         // Initialise the STDAIR service handler
00084         stdair::STDAIR_ServicePtr_T lSTDAIR_Service_ptr =
00085             initStdAirService (iLogParams, iDBParams);
00086
00087         // Initialise the service context
00088         initServiceContext (iCRSCode);
00089
00090         // Add the StdAir service context to the SIMCRS service context
00091         // \note SIMCRS owns the STDAIR service resources here.
00092         const bool ownStdairService = true;
00093         addStdAirService (lSTDAIR_Service_ptr, ownStdairService);
00094
00095         // Initialise the SIMFQT service.
00096         initSIMFQTService();
00097
00098         // Initialise the AIRSCHED service.
00099         initAIRSCHEDService();
00100
00101         // Initialise the AIRINV service.
00102         initAIRINVService();
00103
00104         // Initialise the (remaining of the) context
00105         initSimcrsService();
00106     }
00107
00108     // //////////////////////////////////////
00109     SIMCRS_Service::~SIMCRS_Service (stdair::STDAIR_ServicePtr_T ioSTDAIR_Service_ptr,
00110                                     const CRSCode_T& iCRSCode)
00111         : _simcrsServiceContext (NULL) {
00112
00113         // Initialise the service context
00114         initServiceContext (iCRSCode);
00115
00116         // Store the STDAIR service object within the (AIRINV) service context
00117         // \note AirInv does not own the STDAIR service resources here.
00118         const bool doesNotOwnStdairService = false;
00119

```

```

00120     addStdAirService (ioSTDAIR_Service_ptr, doesNotOwnStdairService);
00121
00122     // Initialise the SIMFQT service.
00123     initSIMFQTService();
00124
00125     // Initialise the AIRSCHED service.
00126     initAIRSCHEDService();
00127
00128     // Initialise the AIRINV service.
00129     initAIRINVService();
00130
00131     // Initialise the (remaining of the) context
00132     initSimcrsService();
00133 }
00134
00135 // //////////////////////////////////////
00136 SIMCRS_Service::~SIMCRS_Service() {
00137     // Delete/Clean all the objects from memory
00138     finalise();
00139 }
00140
00141 // //////////////////////////////////////
00142 void SIMCRS_Service::finalise() {
00143     assert (_simcrsServiceContext != NULL);
00144     // Reset the (Boost.)Smart pointer pointing on the STDAIR_Service object.
00145     _simcrsServiceContext->reset();
00146 }
00147
00148 // //////////////////////////////////////
00149 void SIMCRS_Service::initServiceContext (const CRSCode_T& iCRSCode)
00150 {
00151     // Initialise the service context
00152     SIMCRS_ServiceContext& lSIMCRS_ServiceContext =
00153         FacSimcrsServiceContext::instance().
00154         create (iCRSCode);
00155     _simcrsServiceContext = &lSIMCRS_ServiceContext;
00156 }
00157
00158 // //////////////////////////////////////
00159 void SIMCRS_Service::
00160 addStdAirService (stdair::STDAIR_ServicePtr_T ioSTDAIR_Service_ptr,
00161                  const bool iOwnStdairService) {
00162
00163     // Retrieve the SimCRS service context
00164     assert (_simcrsServiceContext != NULL);
00165     SIMCRS_ServiceContext& lSIMCRS_ServiceContext = *_simcrsServiceContext;
00166
00167     // Store the StdAir service object within the (SimCRS) service context
00168     lSIMCRS_ServiceContext.setSTDAIR_Service (ioSTDAIR_Service_ptr,
00169                                              iOwnStdairService);
00170 }
00171
00172 // //////////////////////////////////////
00173 stdair::STDAIR_ServicePtr_T SIMCRS_Service::
00174 initStdAirService (const stdair::BasLogParams& iLogParams) {
00175
00176     stdair::STDAIR_ServicePtr_T lSTDAIR_Service_ptr =
00177         boost::make_shared<stdair::STDAIR_Service> (iLogParams);
00178
00179     return lSTDAIR_Service_ptr;
00180 }
00181
00182 // //////////////////////////////////////
00183 stdair::STDAIR_ServicePtr_T SIMCRS_Service::
00184 initStdAirService (const stdair::BasLogParams& iLogParams,
00185                   const stdair::BasDBParams& iDBParams) {
00186
00187     stdair::STDAIR_ServicePtr_T lSTDAIR_Service_ptr =
00188         boost::make_shared<stdair::STDAIR_Service> (iLogParams, iDBParams);
00189
00190     return lSTDAIR_Service_ptr;
00191 }
00192
00193 // //////////////////////////////////////
00194 void SIMCRS_Service::initAIRSCHEDService() {
00195
00196     // Retrieve the SimCRS service context
00197     assert (_simcrsServiceContext != NULL);
00198     SIMCRS_ServiceContext& lSIMCRS_ServiceContext = *_simcrsServiceContext;
00199
00200     // Retrieve the StdAir service context
00201     stdair::STDAIR_ServicePtr_T lSTDAIR_Service_ptr =
00202         lSIMCRS_ServiceContext.getSTDAIR_ServicePtr();
00203
00204     AIRSCHED::AIRSCHED_ServicePtr_T lAIRSCHED_Service_ptr =
00205         boost::make_shared<AIRSCHED::AIRSCHED_Service> (lSTDAIR_Service_ptr);
00206 }
00207
00208
00209
00210
00211
00212
00213
00214
00215
00216
00217
00218
00219
00220
00221
00222
00223
00224
00225

```

```

00226 // Store the AIRSCHED service object within the (SimCRS) service context
00227 lSIMCRS_ServiceContext.setAIRSCHED_Service (lAIRSCHED_Service_ptr);
00228 }
00229
00230 // //////////////////////////////////////
00231 void SIMCRS_Service::initSIMFQTService() {
00232
00233 // Retrieve the SimCRS service context
00234 assert (_simcrsServiceContext != NULL);
00235 SIMCRS_ServiceContext& lSIMCRS_ServiceContext = *_simcrsServiceContext;
00236
00237 // Retrieve the StdAir service context
00238 stdair::STDAIR_ServicePtr_T lSTDAIR_Service_ptr =
00239     lSIMCRS_ServiceContext.getSTDAIR_ServicePtr();
00240
00241 SIMFQT::SIMFQT_ServicePtr_T lSIMFQT_Service_ptr =
00242     boost::make_shared<SIMFQT::SIMFQT_Service> (lSTDAIR_Service_ptr);
00243
00244 // Store the SIMFQT service object within the (SimCRS) service context
00245 lSIMCRS_ServiceContext.setSIMFQT_Service (lSIMFQT_Service_ptr);
00246 }
00247
00248 // //////////////////////////////////////
00249 void SIMCRS_Service::initAIRINVService() {
00250
00251 // Retrieve the SimCRS service context
00252 assert (_simcrsServiceContext != NULL);
00253 SIMCRS_ServiceContext& lSIMCRS_ServiceContext = *_simcrsServiceContext;
00254
00255 // Retrieve the StdAir service context
00256 stdair::STDAIR_ServicePtr_T lSTDAIR_Service_ptr =
00257     lSIMCRS_ServiceContext.getSTDAIR_ServicePtr();
00258
00259 AIRINV::AIRINV_Master_ServicePtr_T lAIRINV_Service_ptr =
00260     boost::make_shared<AIRINV::AIRINV_Master_Service> (lSTDAIR_Service_ptr);
00261
00262 // Store the AIRINV service object within the (SimCRS) service context
00263 lSIMCRS_ServiceContext.setAIRINV_Service (lAIRINV_Service_ptr);
00264 }
00265
00266 // //////////////////////////////////////
00267 void SIMCRS_Service::initSimcrsService() {
00268 // Do nothing at this stage. A sample BOM tree may be built by
00269 // calling the buildSampleBom() method
00270 }
00271
00272 // //////////////////////////////////////
00273 void SIMCRS_Service::
00274 parseAndLoad (const stdair::Filename_T& iScheduleInputFilename,
00275               const stdair::Filename_T& iODInputFilename,
00276               const AIRRAC::YieldFilePath& iYieldInputFilepath,
00277               const SIMFQT::FareFilePath& iFareInputFilepath) {
00278
00279 // Retrieve the SimCRS service context
00280 assert (_simcrsServiceContext != NULL);
00281 SIMCRS_ServiceContext& lSIMCRS_ServiceContext = *
00282     _simcrsServiceContext;
00283
00284 AIRSCHED::AIRSCHED_Service& lAIRSCHED_Service =
00285     lSIMCRS_ServiceContext.getAIRSCHED_Service();
00286 lAIRSCHED_Service.parseAndLoad (iScheduleInputFilename);
00287
00288 AIRINV::AIRINV_Master_Service& lAIRINV_Service =
00289     lSIMCRS_ServiceContext.getAIRINV_Service();
00290 lAIRINV_Service.parseAndLoad (iScheduleInputFilename, iODInputFilename,
00291                               iYieldInputFilepath);
00292
00293 SIMFQT::SIMFQT_Service& lSIMFQT_Service =
00294     lSIMCRS_ServiceContext.getSIMFQT_Service();
00295 lSIMFQT_Service.parseAndLoad (iFareInputFilepath);
00296 }
00297
00298 // //////////////////////////////////////
00299 void SIMCRS_Service::buildSampleBom() {
00300
00301 // Retrieve the SimCRS service context
00302 if (_simcrsServiceContext == NULL) {
00303     throw stdair::NonInitialisedServiceException ("The SimCRS service "
00304                                                    "has not been initialised");
00305 }
00306
00307 assert (_simcrsServiceContext != NULL);
00308
00309 // Retrieve the SimCRS service context and whether it owns the Stdair
00310 // service
00311 SIMCRS_ServiceContext& lSIMCRS_ServiceContext = *
00312     _simcrsServiceContext;

```

```

00338     const bool doesOwnStdairService =
00339         lSIMCRS_ServiceContext.getOwnStdairServiceFlag();
00340
00341     // Retrieve the StdAir service object from the (SimCRS) service context
00342     stdair::STDAIR_Service& lSTDAIR_Service =
00343         lSIMCRS_ServiceContext.getSTDAIR_Service();
00344
00345     if (doesOwnStdairService == true) {
00346         //
00347         lSTDAIR_Service.buildSampleBom();
00348     }
00349
00350     AIRSCHED::AIRSCHED_Service& lAIRSCHED_Service =
00351         lSIMCRS_ServiceContext.getAIRSCHED_Service();
00352     lAIRSCHED_Service.buildSampleBom();
00353
00354     AIRINV::AIRINV_Master_Service& lAIRINV_Service =
00355         lSIMCRS_ServiceContext.getAIRINV_Service();
00356     lAIRINV_Service.buildSampleBom();
00357
00358     SIMFQT::SIMFQT_Service& lSIMFQT_Service =
00359         lSIMCRS_ServiceContext.getSIMFQT_Service();
00360     lSIMFQT_Service.buildSampleBom();
00361 }
00362
00363 // //////////////////////////////////////
00364 void SIMCRS_Service::
00365 buildSampleTravelSolutions(
00366     stdair::TravelSolutionList_T& ioTravelSolutionList){
00367
00368     // Retrieve the SimCRS service context
00369     if (_simcrsServiceContext == NULL) {
00370         throw stdair::NonInitialisedServiceException ("The SimCRS service "
00371             "has not been initialised")
00372     };
00373
00374     assert (_simcrsServiceContext != NULL);
00375
00376     // Retrieve the StdAir service object from the (SimCRS) service context
00377     SIMCRS_ServiceContext& lSIMCRS_ServiceContext = *
00378         _simcrsServiceContext;
00379     stdair::STDAIR_Service& lSTDAIR_Service =
00380         lSIMCRS_ServiceContext.getSTDAIR_Service();
00381
00382     // Delegate the BOM building to the dedicated service
00383     lSTDAIR_Service.buildSampleTravelSolutions (ioTravelSolutionList);
00384 }
00385
00386 // //////////////////////////////////////
00387 stdair::BookingRequestStruct SIMCRS_Service::
00388 buildSampleBookingRequest (const bool isForCRS) {
00389
00390     // Retrieve the SimCRS service context
00391     if (_simcrsServiceContext == NULL) {
00392         throw stdair::NonInitialisedServiceException ("The SimCRS service "
00393             "has not been initialised")
00394     };
00395
00396     assert (_simcrsServiceContext != NULL);
00397
00398     // Retrieve the StdAir service object from the (SimCRS) service context
00399     SIMCRS_ServiceContext& lSIMCRS_ServiceContext = *
00400         _simcrsServiceContext;
00401     stdair::STDAIR_Service& lSTDAIR_Service =
00402         lSIMCRS_ServiceContext.getSTDAIR_Service();
00403
00404     // Delegate the BOM building to the dedicated service
00405     return lSTDAIR_Service.buildSampleBookingRequest (isForCRS);
00406 }
00407
00408 // //////////////////////////////////////
00409 std::string SIMCRS_Service::
00410 jsonExport (const stdair::AirlineCode_T& iAirlineCode,
00411             const stdair::FlightNumber_T& iFlightNumber,
00412             const stdair::Date_T& iDepartureDate) const {
00413
00414     // Retrieve the SimCRS service context
00415     if (_simcrsServiceContext == NULL) {
00416         throw stdair::NonInitialisedServiceException ("The SimCRS service "
00417             "has not been initialised")
00418     };
00419
00420     assert (_simcrsServiceContext != NULL);
00421
00422     // Retrieve the StdAir service object from the (SimCRS) service context
00423     SIMCRS_ServiceContext& lSIMCRS_ServiceContext = *
00424 
```

```

_simcrsServiceContext;
00447     stdair::STDAIR_Service& lSTDAIR_Service =
00448         lSIMCRS_ServiceContext.getSTDAIR_Service();
00449
00450     // Delegate the JSON export to the dedicated service
00451     return lSTDAIR_Service.jsonExport (iAirlineCode, iFlightNumber,
00452         iDepartureDate);
00453 }
00454
00455 // //////////////////////////////////////
00456 void SIMCRS_Service::
00457     initSnapshotAndRMEvents (const stdair::Date_T&
00458 iStartDate,
00459                             const stdair::Date_T& iEndDate) {
00459
00460     // Retrieve the SimCRS service context
00461     if (_simcrsServiceContext == NULL) {
00462         throw stdair::NonInitialisedServiceException ("The SimCRS service has "
00463             "not been initialised");
00464     }
00465     assert (_simcrsServiceContext != NULL);
00466     SIMCRS_ServiceContext& lSIMCRS_ServiceContext = *
00467         _simcrsServiceContext;
00467
00468     // Retrieve the AIRINV Master service.
00469     AIRINV::AIRINV_Master_Service& lAIRINV_Master_Service =
00470         lSIMCRS_ServiceContext.getAIRINV_Service();
00471
00472     lAIRINV_Master_Service.initSnapshotAndRMEvents (iStartDate, iEndDate);
00473 }
00474
00475 // //////////////////////////////////////
00476 std::string SIMCRS_Service::csvDisplay() const {
00477
00478     // Retrieve the SimCRS service context
00479     if (_simcrsServiceContext == NULL) {
00480         throw stdair::NonInitialisedServiceException ("The SimCRS service "
00481             "has not been initialised")
00482     }
00482
00483     assert (_simcrsServiceContext != NULL);
00484
00485     // Retrieve the StdAir service object from the (SimCRS) service context
00486     SIMCRS_ServiceContext& lSIMCRS_ServiceContext = *
00487         _simcrsServiceContext;
00487     stdair::STDAIR_Service& lSTDAIR_Service =
00488         lSIMCRS_ServiceContext.getSTDAIR_Service();
00489
00490     // Delegate the BOM building to the dedicated service
00491     return lSTDAIR_Service.csvDisplay();
00492 }
00493
00494 // //////////////////////////////////////
00495 std::string SIMCRS_Service::
00496     csvDisplay (const stdair::TravelSolutionList_T&
00497 ioTravelSolutionList) const {
00497
00498     // Retrieve the SimCRS service context
00499     if (_simcrsServiceContext == NULL) {
00500         throw stdair::NonInitialisedServiceException ("The SimCRS service "
00501             "has not been initialised")
00502     }
00502
00503     assert (_simcrsServiceContext != NULL);
00504
00505     // Retrieve the StdAir service object from the (SimCRS) service context
00506     SIMCRS_ServiceContext& lSIMCRS_ServiceContext = *
00507         _simcrsServiceContext;
00507     stdair::STDAIR_Service& lSTDAIR_Service =
00508         lSIMCRS_ServiceContext.getSTDAIR_Service();
00509
00510     // Delegate the BOM building to the dedicated service
00511     return lSTDAIR_Service.csvDisplay (ioTravelSolutionList);
00512 }
00513
00514 // //////////////////////////////////////
00515 stdair::TravelSolutionList_T SIMCRS_Service::
00516     calculateSegmentPathList (const
00517 stdair::BookingRequestStruct& iBookingRequest){
00517
00518     // Retrieve the SimCRS service context
00519     if (_simcrsServiceContext == NULL) {
00520         throw stdair::NonInitialisedServiceException ("The SimCRS service "
00521             "has not been initialised")
00522     }
00522
00523     assert (_simcrsServiceContext != NULL);

```



```

00524
00525     SIMCRS_ServiceContext& lSIMCRS_ServiceContext = *
        _simcrsServiceContext;
00526
00527     stdair::TravelSolutionList_T oTravelSolutionList;
00528
00529     // Get a reference on the AIRSCHED service handler
00530     AIRSCHED::AIRSCHED_Service& lAIRSCHED_Service =
00531         lSIMCRS_ServiceContext.getAIRSCHED_Service();
00532
00533     // Delegate the booking to the dedicated service
00534     stdair::BasChronometer lTravelSolutionRetrievingChronometer;
00535     lTravelSolutionRetrievingChronometer.start();
00536
00537     lAIRSCHED_Service.buildSegmentPathList (oTravelSolutionList,
00538                                             iBookingRequest);
00539
00540     // DEBUG
00541     const double lSegmentPathRetrievingMeasure =
00542         lTravelSolutionRetrievingChronometer.elapsed();
00543     STDAIR_LOG_DEBUG ("Travel solution retrieving: "
00544                     << lSegmentPathRetrievingMeasure << " - "
00545                     << lSIMCRS_ServiceContext.display());
00546
00547     return oTravelSolutionList;
00548 }
00549
00550 // //////////////////////////////////////
00551 void SIMCRS_Service::
00552 fareQuote (const stdair::BookingRequestStruct& iBookingRequest,
00553           stdair::TravelSolutionList_T& ioTravelSolutionList) {
00554
00555     // Retrieve the SimCRS service context
00556     if (_simcrsServiceContext == NULL) {
00557         throw stdair::NonInitialisedServiceException ("The SimCRS service has "
00558                                                     "not been initialised");
00559     }
00560     assert (_simcrsServiceContext != NULL);
00561
00562     SIMCRS_ServiceContext& lSIMCRS_ServiceContext = *
        _simcrsServiceContext;
00563
00564     // Get a reference on the SIMFQT service handler
00565     SIMFQT::SIMFQT_Service& lSIMFQT_Service =
00566         lSIMCRS_ServiceContext.getSIMFQT_Service();
00567
00568     // Delegate the action to the dedicated command
00569     stdair::BasChronometer lFareQuoteRetrievalChronometer;
00570     lFareQuoteRetrievalChronometer.start();
00571
00572     lSIMFQT_Service.quotePrices (iBookingRequest, ioTravelSolutionList);
00573
00574     // DEBUG
00575     const double lFareQuoteRetrievalMeasure =
00576         lFareQuoteRetrievalChronometer.elapsed();
00577     STDAIR_LOG_DEBUG ("Fare Quote retrieving: " << lFareQuoteRetrievalMeasure
00578                     << " - " << lSIMCRS_ServiceContext.display());
00579 }
00580
00581 // //////////////////////////////////////
00582 void SIMCRS_Service::
00583 calculateAvailability (stdair::TravelSolutionList_T&
00584 ioTravelSolutionList,
00585                      const stdair::PartnershipTechnique&
00586 iPartnershipTechnique) {
00587
00588     // Retrieve the SimCRS service context
00589     if (_simcrsServiceContext == NULL) {
00590         throw stdair::NonInitialisedServiceException ("The SimCRS service has "
00591                                                     "not been initialised");
00592     }
00593     assert (_simcrsServiceContext != NULL);
00594
00595     SIMCRS_ServiceContext& lSIMCRS_ServiceContext = *
        _simcrsServiceContext;
00596
00597     // Retrieve the CRS code
00598     //const CRSCode_T& lCRSCode = lSIMCRS_ServiceContext.getCRSCode();
00599
00600     // Retrieve the AIRINV Master service.
00601     AIRINV::AIRINV_Master_Service& lAIRINV_Master_Service =
00602         lSIMCRS_ServiceContext.getAIRINV_Service();
00603
00604     // Delegate the availability retrieval to the dedicated command
00605     stdair::BasChronometer lAvlChronometer;
00606     lAvlChronometer.start();

```

```

00606     DistributionManager::calculateAvailability
00607     (lAIRINV_Master_Service,
00608                                     ioTravelSolutionList,
00609                                     iPartnershipTechnique);
00610     // DEBUG
00611     const double lAvlMeasure = lAvlChronometer.elapsed();
00612     STDAIR_LOG_DEBUG ("Availability retrieval: " << lAvlMeasure << " - "
00613                     << lSIMCRS_ServiceContext.display());
00614 }
00615
00616 // //////////////////////////////////////
00617 bool SIMCRS_Service::
00618 sell (const stdair::TravelSolutionStruct& iTravelSolution,
00619      const stdair::PartySize_T& iPartySize) {
00620     bool hasSaleBeenSuccessful = false;
00621
00622     // Retrieve the SimCRS service context
00623     if (_simcrsServiceContext == NULL) {
00624         throw stdair::NonInitialisedServiceException ("The SimCRS service has "
00625                                                     "not been initialised");
00626     }
00627     assert (_simcrsServiceContext != NULL);
00628
00629     SIMCRS_ServiceContext& lSIMCRS_ServiceContext = *
00630     _simcrsServiceContext;
00631
00632     // Retrieve the CRS code
00633     //const CRSCode_T& lCRSCode = lSIMCRS_ServiceContext.getCRSCode();
00634
00635     // Retrieve the AIRINV Master service.
00636     AIRINV::AIRINV_Master_Service& lAIRINV_Master_Service =
00637     lSIMCRS_ServiceContext.getAIRINV_Service();
00638
00639     // Delegate the booking to the dedicated command
00640     stdair::BasChronometer lSellChronometer;
00641     lSellChronometer.start();
00642
00643     hasSaleBeenSuccessful = DistributionManager::sell
00644     (lAIRINV_Master_Service,
00645                                     iTravelSolution,
00646                                     iPartySize);
00647
00648     // DEBUG
00649     STDAIR_LOG_DEBUG ("Made a sell of " << iPartySize
00650                     << " persons on the following travel solution: "
00651                     << iTravelSolution.describe()
00652                     << " with the chosen fare option: "
00653                     << iTravelSolution.getChosenFareOption().describe()
00654                     << ". Successful? " << hasSaleBeenSuccessful);
00655
00656     // DEBUG
00657     const double lSellMeasure = lSellChronometer.elapsed();
00658     STDAIR_LOG_DEBUG ("Booking sell: " << lSellMeasure << " - "
00659                     << lSIMCRS_ServiceContext.display());
00660
00661     return hasSaleBeenSuccessful;
00662 }
00663
00664 // //////////////////////////////////////
00665 bool SIMCRS_Service::
00666 playCancellation (const stdair::CancellationStruct&
00667 iCancellation) {
00668     bool hasCancellationBeenSuccessful = false;
00669
00670     // Retrieve the SimCRS service context
00671     if (_simcrsServiceContext == NULL) {
00672         throw stdair::NonInitialisedServiceException ("The SimCRS service has "
00673                                                     "not been initialised");
00674     }
00675     assert (_simcrsServiceContext != NULL);
00676
00677     SIMCRS_ServiceContext& lSIMCRS_ServiceContext = *
00678     _simcrsServiceContext;
00679
00680     // Retrieve the CRS code
00681     //const CRSCode_T& lCRSCode = lSIMCRS_ServiceContext.getCRSCode();
00682
00683     // Retrieve the AIRINV Master service.
00684     AIRINV::AIRINV_Master_Service& lAIRINV_Master_Service =
00685     lSIMCRS_ServiceContext.getAIRINV_Service();
00686
00687     // Delegate the booking to the dedicated command
00688     stdair::BasChronometer lCancellationChronometer;
00689     lCancellationChronometer.start();
00690
00691     hasCancellationBeenSuccessful = DistributionManager::playCancellation
00692     (lAIRINV_Master_Service,
00693                                     iCancellation);
00694
00695     // DEBUG
00696     STDAIR_LOG_DEBUG ("Made a cancellation of " << iCancellation
00697                     << " on the following travel solution: "
00698                     << iCancellation.getTravelSolution().describe()
00699                     << ". Successful? " << hasCancellationBeenSuccessful);
00700
00701     return hasCancellationBeenSuccessful;
00702 }

```

```

00688     hasCancellationBeenSuccessful =
00689         DistributionManager::playCancellation
00690         (lAIRINV_Master_Service,
00691          iCancellation);
00692     // DEBUG
00693     STDAIR_LOG_DEBUG ("Made a cancellation of " << iCancellation.describe());
00694     // DEBUG
00695     const double lCancellationMeasure = lCancellationChronometer.elapsed();
00696     STDAIR_LOG_DEBUG ("Booking cancellation: " << lCancellationMeasure << " - "
00697                     << lSIMCRS_ServiceContext.display());
00699     return hasCancellationBeenSuccessful;
00701 }
00702
00703 ///////////////////////////////////////////////////////////////////
00704 void SIMCRS_Service::takeSnapshots (const
stdair::SnapshotStruct& iSnapshot) {
00705     // Retrieve the SimCRS service context
00706     if (_simcrsServiceContext == NULL) {
00707         throw stdair::NonInitialisedServiceException ("The SimCRS service has "
00708                                                     "not been initialised");
00710     }
00711     assert (_simcrsServiceContext != NULL);
00712     SIMCRS_ServiceContext& lSIMCRS_ServiceContext = *
_simcrsServiceContext;
00713     // Retrieve the AIRINV Master service.
00714     AIRINV::AIRINV_Master_Service& lAIRINV_Master_Service =
00715         lSIMCRS_ServiceContext.getAIRINV_Service();
00717     lAIRINV_Master_Service.takeSnapshots (iSnapshot);
00719 }
00720
00721 ///////////////////////////////////////////////////////////////////
00722 void SIMCRS_Service::
00723 optimise (const stdair::RMEventStruct& iRMEvent,
00724          const stdair::ForecastingMethod& iForecastingMethod,
00725          const stdair::PartnershipTechnique& iPartnershipTechnique) {
00726     // Retrieve the SimCRS service context
00727     if (_simcrsServiceContext == NULL) {
00728         throw stdair::NonInitialisedServiceException ("The SimCRS service has "
00729                                                     "not been initialised");
00731     }
00732     assert (_simcrsServiceContext != NULL);
00733     SIMCRS_ServiceContext& lSIMCRS_ServiceContext = *
_simcrsServiceContext;
00734     // Retrieve the AIRINV Master service.
00735     AIRINV::AIRINV_Master_Service& lAIRINV_Master_Service =
00736         lSIMCRS_ServiceContext.getAIRINV_Service();
00738     lAIRINV_Master_Service.optimise (iRMEvent, iForecastingMethod,
iPartnershipTechnique);
00740 }
00741 }

```

## 23.57 simcrs/service/SIMCRS\_ServiceContext.cpp File Reference

```

#include <cassert>
#include <stdair/STDAIR_Service.hpp>
#include <stdair/service/Logger.hpp>
#include <simcrs/basic/BasConst_SIMCRS_Service.hpp>
#include <simcrs/service/SIMCRS_ServiceContext.hpp>

```

### Namespaces

- namespace [SIMCRS](#)

## 23.58 SIMCRS\_ServiceContext.cpp

```

00001 ///////////////////////////////////////////////////////////////////

```

```

00002 // Import section
00003 // //////////////////////////////////////
00004 // STL
00005 #include <cassert>
00006 // Standard Airline Object Model
00007 #include <stdair/STDAIR_Service.hpp>
00008 #include <stdair/service/Logger.hpp>
00009 // Simcrs
00010 #include <simcrs/basic/BasConst_SIMCRS_Service.hpp>
00011 >
00012 #include <simcrs/service/SIMCRS_ServiceContext.hpp>
00013 >
00012 namespace SIMCRS {
00013 namespace SIMCRS {
00014 // //////////////////////////////////////
00015 SIMCRS_ServiceContext::SIMCRS_ServiceContext ()
00016 : _ownStdairService (false), _CRSCode (DEFAULT_CRSCODE) {
00017 }
00018 // //////////////////////////////////////
00019 SIMCRS_ServiceContext::SIMCRS_ServiceContext (const SIMCRS_ServiceContext&)
00020 : _ownStdairService (false) {
00021 }
00022 // //////////////////////////////////////
00023 SIMCRS_ServiceContext::SIMCRS_ServiceContext (const CRSCode_T&
00024 iCRSCode)
00025 : _CRSCode (iCRSCode) {
00026 }
00027 // //////////////////////////////////////
00028 SIMCRS_ServiceContext::~SIMCRS_ServiceContext() {
00029 }
00030 // //////////////////////////////////////
00031 const std::string SIMCRS_ServiceContext::shortDisplay() const {
00032     std::ostringstream ostr;
00033     ostr << "SIMCRS_ServiceContext [" << _CRSCode
00034     << "] - Owns StdAir service: " << _ownStdairService;
00035     return ostr.str();
00036 }
00037 // //////////////////////////////////////
00038 const std::string SIMCRS_ServiceContext::display() const {
00039     std::ostringstream ostr;
00040     ostr << shortDisplay();
00041     return ostr.str();
00042 }
00043 // //////////////////////////////////////
00044 const std::string SIMCRS_ServiceContext::describe() const {
00045     return shortDisplay();
00046 }
00047 // //////////////////////////////////////
00048 void SIMCRS_ServiceContext::reset() {
00049     if (_ownStdairService == true) {
00050         _stdairService.reset();
00051     }
00052 }
00053 }
00054 }
00055 }
00056 }
00057 }
00058 }
00059 }
00060 }
00061 }

```

## 23.59 simcrs/service/SIMCRS\_ServiceContext.hpp File Reference

```

#include <string>
#include <map>
#include <boost/shared_ptr.hpp>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_service_types.hpp>
#include <airinv/AIRINV_Types.hpp>
#include <airsched/AIRSCHEM_Types.hpp>
#include <simfqt/SIMFQT_Types.hpp>
#include <simcrs/SIMCRS_Types.hpp>
#include <simcrs/service/ServiceAbstract.hpp>

```

## Classes

- class [SIMCRS::SIMCRS\\_ServiceContext](#)  
Class holding the context of the Simcrs services.

## Namespaces

- namespace [SIMCRS](#)

## 23.60 SIMCRS\_ServiceContext.hpp

```

00001 #ifndef __SIMCRS_SVC_SIMCRSSERVICECONTEXT_HPP
00002 #define __SIMCRS_SVC_SIMCRSSERVICECONTEXT_HPP
00003
00004 // //////////////////////////////////////
00005 // Import section
00006 // //////////////////////////////////////
00007 // STL
00008 #include <string>
00009 #include <map>
00010 // Boost
00011 #include <boost/shared_ptr.hpp>
00012 // StdAir
00013 #include <stdair/stdair_basic_types.hpp>
00014 #include <stdair/stdair_service_types.hpp>
00015 // AirInv
00016 #include <airinv/AIRINV_Types.hpp>
00017 // AirSched
00018 #include <airsched/AIRSCHED_Types.hpp>
00019 // SimFQT
00020 #include <simfqt/SIMFQT_Types.hpp>
00021 // SimCRS
00022 #include <simcrs/SIMCRS_Types.hpp>
00023 #include <simcrs/service/ServiceAbstract.hpp>
00024
00025 namespace SIMCRS {
00026
00030     class SIMCRS_ServiceContext : public ServiceAbstract
00031     {
00032     public:
00033         friend class SIMCRS_Service;
00034         friend class FacSimcrsServiceContext;
00035
00036     private:
00037         // ////////////////////////////////////// Getters //////////////////////////////////////
00038         const CRSCode_T& getCRSCode() const {
00039             return _CRSCode;
00040         }
00041
00042         stdair::STDAIR_ServicePtr_T getSTDAIR_ServicePtr() const {
00043             return _stdairService;
00044         }
00045
00046         stdair::STDAIR_Service& getSTDAIR_Service() const {
00047             assert (_stdairService != NULL);
00048             return *_stdairService;
00049         }
00050
00051         const bool getOwnStdairServiceFlag() const {
00052             return _ownStdairService;
00053         }
00054
00055         AIRINV::AIRINV_Master_Service& getAIRINV_Service() const {
00056             assert (_airinvService != NULL);
00057             return *_airinvService;
00058         }
00059
00060         AIRSCHED::AIRSCHED_Service& getAIRSCHED_Service() const {
00061             assert (_airschedService != NULL);
00062             return *_airschedService;
00063         }
00064
00065         SIMFQT::SIMFQT_Service& getSIMFQT_Service() const {
00066             assert (_simfqtService != NULL);
00067             return *_simfqtService;
00068         }
00069
00070     private:
00071         // ////////////////////////////////////// Setters //////////////////////////////////////
00072         void setCRSCode (const CRSCode_T& iCRSCode) {

```

```

00103     _CRSCode = iCRSCode;
00104 }
00105
00109 void setSTDAIR_Service (stdair::STDAIR_ServicePtr_T ioSTDAIR_ServicePtr,
00110                        const bool iOwnStdairService) {
00111     _stdairService = ioSTDAIR_ServicePtr;
00112     _ownStdairService = iOwnStdairService;
00113 }
00114
00118 void setAIRINV_Service (AIRINV::AIRINV_Master_ServicePtr_T ioServicePtr) {
00119     _airinvService = ioServicePtr;
00120 }
00121
00125 void setAIRSCHED_Service (AIRSCHED::AIRSCHED_ServicePtr_T ioServicePtr) {
00126     _airschedService = ioServicePtr;
00127 }
00128
00132 void setSIMFQT_Service (SIMFQT::SIMFQT_ServicePtr_T ioServicePtr) {
00133     _simfqtService = ioServicePtr;
00134 }
00135
00136
00137 private:
00138     // ////////////////////////////////// Display Methods //////////////////////////////////
00142     const std::string shortDisplay() const;
00143
00147     const std::string display() const;
00148
00152     const std::string describe() const;
00153
00154
00155 private:
00157
00160     SIMCRS_ServiceContext (const CRSCode_T& iCRSCode);
00164     SIMCRS_ServiceContext ();
00168     SIMCRS_ServiceContext (const SIMCRS_ServiceContext&);
00169
00173     ~SIMCRS_ServiceContext ();
00174
00178     void reset();
00179
00180
00181 private:
00185     stdair::STDAIR_ServicePtr_T _stdairService;
00186
00190     bool _ownStdairService;
00191
00195     AIRSCHED::AIRSCHED_ServicePtr_T _airschedService;
00196
00200     AIRINV::AIRINV_Master_ServicePtr_T _airinvService;
00201
00205     SIMFQT::SIMFQT_ServicePtr_T _simfqtService;
00206
00207
00208 private:
00209     // ////////////////////////////////// Attributes //////////////////////////////////
00215     CRSCode_T _CRSCode;
00216 };
00217
00218 }
00219 #endif // __SIMCRS_SVC_SIMCRSSERVICECONTEXT_HPP

```

## 23.61 simcrs/SIMCRS\_Service.hpp File Reference

```

#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_service_types.hpp>
#include <stdair/basic/ForecastingMethod.hpp>
#include <stdair/basic/PartnershipTechnique.hpp>
#include <stdair/bom/TravelSolutionTypes.hpp>
#include <simfqt/SIMFQT_Types.hpp>
#include <airrac/AIRRAC_Types.hpp>
#include <simcrs/SIMCRS_Types.hpp>

```

### Classes

- class [SIMCRS::SIMCRS\\_Service](#)

## Namespaces

- namespace `stdair`  
    *Forward declarations.*
- namespace `SIMCRS`

## 23.62 SIMCRS\_Service.hpp

```

00001 #ifndef __SIMCRS_SVC_SIMCRS_SERVICE_HPP
00002 #define __SIMCRS_SVC_SIMCRS_SERVICE_HPP
00003
00004 // //////////////////////////////////////
00005 // Import section
00006 // //////////////////////////////////////
00007 // StdAir
00008 #include <stdair/stdair_basic_types.hpp>
00009 #include <stdair/stdair_service_types.hpp>
00010 #include <stdair/basic/ForecastingMethod.hpp>
00011 #include <stdair/basic/PartnershipTechnique.hpp>
00012 #include <stdair/bom/TravelSolutionTypes.hpp>
00013 // SimFQT
00014 #include <simfqt/SIMFQT_Types.hpp>
00015 // AIRRAC
00016 #include <airrac/AIRRAC_Types.hpp>
00017 // SimCRS
00018 #include <simcrs/SIMCRS_Types.hpp>
00019
00021 namespace stdair {
00022     struct BasLogParams;
00023     struct BasDBParams;
00024     struct BookingRequestStruct;
00025     struct CancellationStruct;
00026     struct SnapshotStruct;
00027     struct RMEventStruct;
00028 }
00029
00030 namespace SIMCRS {
00031
00033     class SIMCRS_ServiceContext;
00034
00035
00039     class SIMCRS_Service {
00040     public:
00041         // ////////////////////////////////// Constructors and Destructors //////////////////////////////////
00058         SIMCRS_Service (const stdair::BasLogParams&, const
stdair::BasDBParams&,
00059                         const CRSCode_T&);
00060
00073         SIMCRS_Service (const stdair::BasLogParams&, const CRSCode_T
&);
00074
00091         SIMCRS_Service (stdair::STDAIR_ServicePtr_T, const CRSCode_T
&);
00092
00093
00105         void parseAndLoad (const stdair::Filename_T&
iScheduleInputFilename,
00106                             const stdair::Filename_T& iODInputFilename,
00107                             const AIRRAC::YieldFilePath& iYieldInputFilepath,
00108                             const SIMFQT::FareFilePath& iFareInputFilepath);
00109
00116         void initSnapshotAndRMEvents (const stdair::Date_T&
iStartDate,
00117                                         const stdair::Date_T& iEndDate);
00118
00122         ~SIMCRS_Service();
00123
00124
00125     public:
00126         // ////////////////////////////////// Business Methods //////////////////////////////////
00131         stdair::TravelSolutionList_T
00132         calculateSegmentPathList (const
stdair::BookingRequestStruct&);
00133
00137         void fareQuote (const stdair::BookingRequestStruct&,
stdair::TravelSolutionList_T&);
00138
00139
00143         void calculateAvailability (
stdair::TravelSolutionList_T&,
00144                                     const stdair::PartnershipTechnique&);
00145
00149         bool sell (const stdair::TravelSolutionStruct&, const

```

```

stdair::PartySize_T&);
00150
00154     void takeSnapshots (const stdair::SnapshotStruct&);
00155
00159     bool playCancellation (const stdair::CancellationStruct&);
00160
00164     void optimise (const stdair::RMEventStruct&,
00165                  const stdair::ForecastingMethod&,
00166                  const stdair::PartnershipTechnique&);
00167
00177     void buildSampleBom ();
00178
00198     void buildSampleTravelSolutions (
stdair::TravelSolutionList_T&);
00199
00230     stdair::BookingRequestStruct
00231     buildSampleBookingRequest (const bool isForCRS =
false);
00232
00233
00234     public:
00235         // //////////// Export support methods ////////////
00246         std::string jsonExport (const stdair::AirlineCode_T&,
00247                                const stdair::FlightNumber_T&,
00248                                const stdair::Date_T& iDepartureDate) const;
00249
00250
00251     public:
00252         // //////////// Display support methods ////////////
00260         std::string csvDisplay() const;
00261
00269         std::string csvDisplay (const stdair::TravelSolutionList_T&)
const;
00270
00271
00272     private:
00273         // ////////// Construction and Destruction helper methods //////////
00277         SIMCRS_Service();
00278
00282         SIMCRS_Service (const SIMCRS_Service&);
00283
00293         stdair::STDAIR_ServicePtr_T initStdAirService (const stdair::BasLogParams&,
00294                                                         const stdair::BasDBParams&);
00295
00305         stdair::STDAIR_ServicePtr_T initStdAirService (const stdair::BasLogParams&)
;
00306
00310         void initAIRSCHEDService();
00311
00315         void initSIMFQTSservice();
00316
00320         void initAIRINVService();
00321
00330         void addStdAirService (stdair::STDAIR_ServicePtr_T,
00331                                const bool iOwnStdairService);
00332
00339         void initServiceContext (const CRSCode_T&);
00340
00345         void initSimcrsService();
00346
00350         void finalise();
00351
00352
00353     private:
00354         // ////////// Service Context //////////
00358         SIMCRS_ServiceContext* _simcrsServiceContext;
00359     };
00360 }
00361 #endif // __SIMCRS_SVC_SIMCRS_SERVICE_HPP

```

## 23.63 simcrs/SIMCRS\_Types.hpp File Reference

```

#include <exception>
#include <string>
#include <boost/shared_ptr.hpp>
#include <stdair/stdair_exceptions.hpp>

```



## Classes

- class [SIMCRS::BookingException](#)
- class [SIMCRS::AvailabilityRetrievalException](#)

## Namespaces

- namespace [SIMCRS](#)

## Typedefs

- typedef std::string [SIMCRS::CRSCode\\_T](#)
- typedef boost::shared\_ptr  
     < SIMCRS\_Service > [SIMCRS::SIMCRS\\_ServicePtr\\_T](#)

## 23.64 SIMCRS\_Types.hpp

```

00001 #ifndef __SIMCRS_SIMCRS_TYPES_HPP
00002 #define __SIMCRS_SIMCRS_TYPES_HPP
00003
00004 // //////////////////////////////////////
00005 // Import section
00006 // //////////////////////////////////////
00007 // STL
00008 #include <exception>
00009 #include <string>
00010 // Boost
00011 #include <boost/shared_ptr.hpp>
00012 // StdAir
00013 #include <stdair/stdair_exceptions.hpp>
00014
00015 namespace SIMCRS {
00016
00017     // Forward declarations
00018     class SIMCRS_Service;
00019
00020
00021     // /////////// Exceptions ///////////
00025     class BookingException : public stdair::RootException {
00026     };
00027
00031     class AvailabilityRetrievalException : public
stdair::RootException {
00032     };
00033
00034
00035     // /////////// Type definitions specific to SimCRS ///////////
00039     typedef std::string CRSCode_T;
00040
00044     typedef boost::shared_ptr<SIMCRS_Service> SIMCRS_ServicePtr_T
;
00045
00046 }
00047 #endif // __SIMCRS_SIMCRS_TYPES_HPP
00048

```

## 23.65 test/simcrs/CRSTestSuite.cpp File Reference

## 23.66 CRSTestSuite.cpp

```

00001
00005 // //////////////////////////////////////
00006 // Import section
00007 // //////////////////////////////////////
00008 // STL
00009 #include <sstream>
00010 #include <fstream>
00011 #include <string>
00012 #include <cmath>
00013 // Boost Unit Test Framework (UTF)
00014 #define BOOST_TEST_DYN_LINK

```

```

00015 #define BOOST_TEST_MAIN
00016 #define BOOST_TEST_MODULE CRSTestSuite
00017 #include <boost/test/unit_test.hpp>
00018 // StdAir
00019 #include <stdair/basic/BasLogParams.hpp>
00020 #include <stdair/basic/BasDBParams.hpp>
00021 #include <stdair/basic/BasFileMgr.hpp>
00022 #include <stdair/bom/TravelSolutionStruct.hpp>
00023 #include <stdair/bom/BookingRequestStruct.hpp>
00024 #include <stdair/service/Logger.hpp>
00025 // SimFQT
00026 #include <simfqt/SIMFQT_Types.hpp>
00027 // SimCRS
00028 #include <simcrs/SIMCRS_Service.hpp>
00029 #include <simcrs/config/simcrs-paths.hpp>
00030
00031 namespace boost_utf = boost::unit_test;
00032
00033 // (Boost) Unit Test XML Report
00034 std::ofstream utfReportStream ("CRSTestSuite_utfresults.xml");
00035
00036 struct UnitTestConfig {
00037     UnitTestConfig() {
00038         boost_utf::unit_test_log.set_stream (utfReportStream);
00039         boost_utf::unit_test_log.set_format (boost_utf::XML);
00040         boost_utf::unit_test_log.set_threshold_level (boost_utf::log_test_units);
00041         //boost_utf::unit_test_log.set_threshold_level
00042         (boost_utf::log_successful_tests);
00043     }
00044
00045     ~UnitTestConfig() {
00046     }
00047 };
00048
00049 // Main: Unit Test Suite
00050 BOOST_GLOBAL_FIXTURE (UnitTestConfig);
00051
00052 // Start the test suite
00053 BOOST_AUTO_TEST_SUITE (master_test_suite)
00054
00055 BOOST_AUTO_TEST_CASE (simcrs_simple_simulation_test) {
00056     // CRS code
00057     const SIMCRS::CRSCode_T lCRSCode ("1P");
00058
00059     // Schedule input filename
00060     const stdair::Filename_T lScheduleInputFilename (STDAIR_SAMPLE_DIR
00061         "/rds01/schedule.csv");
00062
00063     // O&D input filename
00064     const stdair::Filename_T lOnDInputFilename (STDAIR_SAMPLE_DIR
00065         "/ond01.csv");
00066
00067     // Yield input filename
00068     const stdair::Filename_T lYieldInputFilename (STDAIR_SAMPLE_DIR
00069         "/rds01/yield.csv");
00070
00071     // Fare input filename
00072     const stdair::Filename_T lFareInputFilename (STDAIR_SAMPLE_DIR
00073         "/rds01/fare.csv");
00074
00075     // Check that the file path given as input corresponds to an actual file
00076     bool doesExistAndIsReadable =
00077         stdair::BasFileMgr::doesExistAndIsReadable (lScheduleInputFilename);
00078     BOOST_CHECK_MESSAGE (doesExistAndIsReadable == true,
00079         "The '" << lScheduleInputFilename
00080         << "' input file can not be open and read");
00081
00082     // Check that the file path given as input corresponds to an actual file
00083     doesExistAndIsReadable =
00084         stdair::BasFileMgr::doesExistAndIsReadable (lOnDInputFilename);
00085     BOOST_CHECK_MESSAGE (doesExistAndIsReadable == true,
00086         "The '" << lOnDInputFilename
00087         << "' input file can not be open and read");
00088
00089     // Check that the file path given as input corresponds to an actual file
00090     doesExistAndIsReadable =
00091         stdair::BasFileMgr::doesExistAndIsReadable (lYieldInputFilename);
00092     BOOST_CHECK_MESSAGE (doesExistAndIsReadable == true,
00093         "The '" << lYieldInputFilename
00094         << "' input file can not be open and read");
00095
00096     // Check that the file path given as input corresponds to an actual file
00097     doesExistAndIsReadable =
00098         stdair::BasFileMgr::doesExistAndIsReadable (lFareInputFilename);
00099     BOOST_CHECK_MESSAGE (doesExistAndIsReadable == true,
00100         "The '" << lFareInputFilename
00101         << "' input file can not be open and read");
00102 }

```

```

00107     doesExistAndIsReadable =
00108         stdair::BasFileMgr::doesExistAndIsReadable (lFareInputFilename);
00109     BOOST_CHECK_MESSAGE (doesExistAndIsReadable == true,
00110         "The '" << lFareInputFilename
00111         << "' input file can not be open and read");
00112
00113     // Output log File
00114     const stdair::Filename_T lLogFilename ("CRSTestSuite.log");
00115
00116     // Set the log parameters
00117     std::ofstream logOutputFile;
00118     // Open and clean the log outputfile
00119     logOutputFile.open (lLogFilename.c_str());
00120     logOutputFile.clear();
00121
00122     // Initialise the list of classes/buckets
00123     const stdair::BasLogParams lLogParams (stdair::LOG::DEBUG, logOutputFile);
00124     SIMCRS::SIMCRS_Service simcrsService (lLogParams,
00125     lCRSCode);
00126
00127     // Build the BOM tree from parsing input files
00128     const SIMFQT::FareFilePath lFareFilePath (lFareInputFilename);
00129     const AIRRAC::YieldFilePath lYieldFilePath (lYieldInputFilename);
00130     simcrsService.parseAndLoad (lScheduleInputFilename, lOnDInputFilename,
00131     lYieldFilePath, lFareFilePath);
00132
00133     // Create an empty booking request structure
00134     // TODO: fill the booking request structure from the input parameters
00135     const stdair::AirportCode_T lOrigin ("SIN");
00136     const stdair::AirportCode_T lDestination ("BKK");
00137     const stdair::AirportCode_T lPOS ("SIN");
00138     const stdair::Date_T lPreferredDepartureDate(2011, boost::gregorian::Jan, 31)
00139 ;
00140     const stdair::Date_T lRequestDate (2011, boost::gregorian::Jan, 22);
00141     const stdair::Duration_T lRequestTime (boost::posix_time::hours(10));
00142     const stdair::Date_T lRequestDateTime (lRequestDate, lRequestTime);
00143     const stdair::CabinCode_T lPreferredCabin ("Eco");
00144     const stdair::PartySize_T lPartySize (3);
00145     const stdair::ChannelLabel_T lChannel ("IN");
00146     const stdair::TripType_T lTripType ("RI");
00147     const stdair::DayDuration_T lStayDuration (7);
00148     const stdair::FrequentFlyer_T lFrequentFlyerType ("M");
00149     const stdair::Duration_T lPreferredDepartureTime (boost::posix_time::hours(10)
00150 ));
00151     const stdair::WTP_T lWTP (1000.0);
00152     const stdair::PriceValue_T lValueOfTime (100.0);
00153     const stdair::BookingRequestStruct lBookingRequest (lOrigin, lDestination,
00154     lPOS,
00155     lPreferredDepartureDate,
00156     lRequestDateTime,
00157     lPreferredCabin,
00158     lPartySize, lChannel,
00159     lTripType, lStayDuration,
00160     lFrequentFlyerType,
00161     lPreferredDepartureTime,
00162     lWTP, lValueOfTime);
00163     stdair::TravelSolutionList_T lTravelSolutionList =
00164     simcrsService.calculateSegmentPathList (lBookingRequest);
00165
00166     // Price the travel solution
00167     simcrsService.fareQuote (lBookingRequest, lTravelSolutionList);
00168
00169     //
00170     const unsigned int lNbOfTravelSolutions = lTravelSolutionList.size();
00171
00172     // \todo change the expected number of travel solutions to the actual number
00173     const unsigned int lExpectedNbOfTravelSolutions = 1;
00174
00175     // DEBUG
00176     std::ostringstream oMessageKeptTS;
00177     oMessageKeptTS << "The number of travel solutions for the booking request '"
00178     << lBookingRequest.describe() << "' is actually "
00179     << lNbOfTravelSolutions << ". That number is expected to be "
00180     << lExpectedNbOfTravelSolutions << ".";
00181     STDAIR_LOG_DEBUG (oMessageKeptTS.str());
00182
00183     BOOST_CHECK_EQUAL (lNbOfTravelSolutions, lExpectedNbOfTravelSolutions);
00184
00185     BOOST_CHECK_MESSAGE (lNbOfTravelSolutions == lExpectedNbOfTravelSolutions,
00186     oMessageKeptTS.str());
00187
00188     stdair::TravelSolutionStruct& lTravelSolution = lTravelSolutionList.front();
00189
00190     const stdair::FareOptionList_T& lFareOptionList =
00191     lTravelSolution.getFareOptionList();
00192
00193     stdair::FareOptionStruct lFareOption = lFareOptionList.front();

```

```
00207 lTravelSolution.setChosenFareOption (lFareOption);
00208
00214 const unsigned int lExpectedPrice = 400;
00215
00216 // DEBUG
00217 std::ostream oMessageKeptFare;
00218 oMessageKeptFare
00219     << "The price given by the fare quoter for the booking request: '"
00220     << lBookingRequest.describe() << "' and travel solution: '"
00221     << lTravelSolution.describe() << "' is actually " << lFareOption.getFare()
00222     << " Euros. It is expected to be " << lExpectedPrice << " Euros.";
00223 STDAIR_LOG_DEBUG (oMessageKeptFare.str());
00224
00225 BOOST_CHECK_EQUAL (std::floor (lFareOption.getFare() + 0.5), lExpectedPrice);
00226
00227 BOOST_CHECK_MESSAGE (std::floor (lFareOption.getFare() + 0.5)
00228                     == lExpectedPrice, oMessageKeptFare.str());
00229
00237 // DEBUG
00238 STDAIR_LOG_DEBUG ("A booking will now (attempted to) be made on the "
00239                 "travel solution '" << lTravelSolution.describe()
00240                 << "', for a party size of " << lPartySize << ".");
00241
00242 const bool isSellSuccessful =
00243     simcrsService.sell (lTravelSolution, lPartySize);
00244 //BOOST_CHECK_NO_THROW ();
00245
00246 // DEBUG
00247 std::ostream oMessageSell;
00248 const std::string isSellSuccessfulStr = (isSellSuccessful == true)?"Yes":"No"
;
00249 oMessageSell << "Was the sell successful? Answer: " << isSellSuccessfulStr;
00250 STDAIR_LOG_DEBUG (oMessageSell.str());
00251
00252 BOOST_CHECK_EQUAL (isSellSuccessful, true);
00253
00254 BOOST_CHECK_MESSAGE (isSellSuccessful == true, oMessageSell.str());
00255
00256 // Close the log file
00257 logOutputFile.close();
00258 }
00259
00260 // End the test suite
00261 BOOST_AUTO_TEST_SUITE_END()
00262
00263
```

## Index

- ~BomAbstract
  - SIMCRS::BomAbstract, [87](#)
- ~FacBomAbstract
  - SIMCRS::FacBomAbstract, [89](#)
- ~FacServiceAbstract
  - SIMCRS::FacServiceAbstract, [91](#)
- ~FacSimcrsServiceContext
  - SIMCRS::FacSimcrsServiceContext, [93](#)
- ~FacSupervisor
  - SIMCRS::FacSupervisor, [95](#)
- ~SIMCRS\_Service
  - SIMCRS::SIMCRS\_Service, [100](#)
- ~ServiceAbstract
  - SIMCRS::ServiceAbstract, [98](#)
- \_pool
  - SIMCRS::FacBomAbstract, [90](#)
  - SIMCRS::FacServiceAbstract, [92](#)
  - SIMCRS::FacSimcrsServiceContext, [94](#)
- AIRINV, [84](#)
- BINDIR
  - simcrs-paths.hpp, [119](#)
  - simcrs-paths.hpp.in, [121](#)
- BomAbstract
  - SIMCRS::BomAbstract, [87](#)
- BomAbstract.hpp
  - operator<<, [115](#)
  - operator>>, [115](#)
- BomFactoryPool\_T
  - SIMCRS::FacSupervisor, [95](#)
- BomPool\_T
  - SIMCRS::FacBomAbstract, [89](#)
- buildSampleBom
  - SIMCRS::SIMCRS\_Service, [101](#)
- buildSampleBookingRequest
  - SIMCRS::SIMCRS\_Service, [102](#)
- buildSampleTravelSolutions
  - SIMCRS::SIMCRS\_Service, [101](#)
- CRSCode\_T
  - SIMCRS, [85](#)
- calculateAvailability
  - SIMCRS::SIMCRS\_Service, [101](#)
- calculateSegmentPathList
  - SIMCRS::SIMCRS\_Service, [100](#)
- clean
  - SIMCRS::FacServiceAbstract, [92](#)
  - SIMCRS::FacSimcrsServiceContext, [94](#)
- cleanBomLayer
  - SIMCRS::FacSupervisor, [96](#)
- cleanFactory
  - SIMCRS::FacSupervisor, [96](#)
- cleanServiceLayer
  - SIMCRS::FacSupervisor, [96](#)
- create
  - SIMCRS::FacSimcrsServiceContext, [94](#)
- csvDisplay
  - SIMCRS::SIMCRS\_Service, [103](#)
- DATADIR
  - simcrs-paths.hpp, [120](#)
  - simcrs-paths.hpp.in, [122](#)
- DATAROOTDIR
  - simcrs-paths.hpp, [120](#)
  - simcrs-paths.hpp.in, [122](#)
- DEFAULT\_CRS\_CODE
  - SIMCRS, [85](#)
- DOCDIR
  - simcrs-paths.hpp, [120](#)
  - simcrs-paths.hpp.in, [122](#)
- describeKey
  - SIMCRS::BomAbstract, [87](#)
- describeShortKey
  - SIMCRS::BomAbstract, [87](#)
- doc/local/authors.doc, [105](#)
- doc/local/codingrules.doc, [105](#)
- doc/local/copyright.doc, [105](#)
- doc/local/documentation.doc, [105](#)
- doc/local/features.doc, [105](#)
- doc/local/help\_wanted.doc, [105](#)
- doc/local/howto\_release.doc, [105](#)
- doc/local/index.doc, [105](#)
- doc/local/installation.doc, [105](#)
- doc/local/linking.doc, [105](#)
- doc/local/test.doc, [105](#)
- doc/local/users\_guide.doc, [105](#)
- doc/local/verification.doc, [105](#)
- doc/tutorial/tutorial.doc, [105](#)
- EXEC\_PREFIX
  - simcrs-paths.hpp, [119](#)
  - simcrs-paths.hpp.in, [121](#)
- FacBomAbstract
  - SIMCRS::BomAbstract, [87](#)
  - SIMCRS::FacBomAbstract, [89](#)
- FacServiceAbstract
  - SIMCRS::FacServiceAbstract, [91](#)
- FacSimcrsServiceContext
  - SIMCRS::FacSimcrsServiceContext, [93](#)
  - SIMCRS::SIMCRS\_ServiceContext, [105](#)
- FacSupervisor
  - SIMCRS::FacBomAbstract, [90](#)
  - SIMCRS::FacSupervisor, [95](#)
- fareQuote
  - SIMCRS::SIMCRS\_Service, [101](#)
- fromStream
  - SIMCRS::BomAbstract, [87](#)
  - SIMCRS::ServiceAbstract, [98](#)
  - SIMCRS::SIMCRS\_ServiceContext, [104](#)

- getID
  - SIMCRS::FacBomAbstract, 90
- getIDString
  - SIMCRS::FacBomAbstract, 90
- HTMLDIR
  - simcrs-paths.hpp, 120
  - simcrs-paths.hpp.in, 122
- INCLUDEDIR
  - simcrs-paths.hpp, 120
  - simcrs-paths.hpp.in, 122
- INFODIR
  - simcrs-paths.hpp, 120
  - simcrs-paths.hpp.in, 122
- initSnapshotAndRMEvents
  - SIMCRS::SIMCRS\_Service, 100
- instance
  - SIMCRS::FacSimcrsServiceContext, 93
  - SIMCRS::FacSupervisor, 96
- jsonExport
  - SIMCRS::SIMCRS\_Service, 103
- LIBDIR
  - simcrs-paths.hpp, 119
  - simcrs-paths.hpp.in, 121
- LIBEXECDIR
  - simcrs-paths.hpp, 119
  - simcrs-paths.hpp.in, 121
- MANDIR
  - simcrs-paths.hpp, 120
  - simcrs-paths.hpp.in, 122
- main
  - simcrs.cpp, 109
- operator<<
  - BomAbstract.hpp, 115
  - ServiceAbstract.hpp, 131
  - simcrs.cpp, 108
- operator>>
  - BomAbstract.hpp, 115
  - ServiceAbstract.hpp, 131
- optimise
  - SIMCRS::SIMCRS\_Service, 101
- PACKAGE
  - simcrs-paths.hpp, 119
  - simcrs-paths.hpp.in, 121
- PACKAGE\_NAME
  - simcrs-paths.hpp, 119
  - simcrs-paths.hpp.in, 121
- PACKAGE\_VERSION
  - simcrs-paths.hpp, 119
  - simcrs-paths.hpp.in, 121
- PDFDIR
  - simcrs-paths.hpp, 120
  - simcrs-paths.hpp.in, 122
- PREFIXDIR
  - simcrs-paths.hpp, 119
  - simcrs-paths.hpp.in, 121
- parseAndLoad
  - SIMCRS::SIMCRS\_Service, 100
- playCancellation
  - SIMCRS::SIMCRS\_Service, 101
- readConfiguration
  - simcrs.cpp, 108
- registerBomFactory
  - SIMCRS::FacSupervisor, 96
- registerServiceFactory
  - SIMCRS::FacSupervisor, 96
- RootException, 97
- SBINDIR
  - simcrs-paths.hpp, 119
  - simcrs-paths.hpp.in, 122
- SIMCRS, 84
  - CRSCode\_T, 85
  - SIMCRS\_ServicePtr\_T, 85
- SIMCRS::AvailabilityRetrievalException, 86
- SIMCRS::BomAbstract, 86
  - ~BomAbstract, 87
  - BomAbstract, 87
  - describeKey, 87
  - describeShortKey, 87
  - FacBomAbstract, 87
  - fromStream, 87
  - toStream, 87
  - toString, 87
- SIMCRS::BookingException, 88
- SIMCRS::DistributionManager, 88
  - SIMCRS\_Service, 88
- SIMCRS::FacBomAbstract, 88
  - ~FacBomAbstract, 89
  - \_pool, 90
  - BomPool\_T, 89
  - FacBomAbstract, 89
  - FacSupervisor, 90
  - getID, 90
  - getIDString, 90
- SIMCRS::FacServiceAbstract, 91
  - ~FacServiceAbstract, 91
  - \_pool, 92
  - clean, 92
  - FacServiceAbstract, 91
  - ServicePool\_T, 91
- SIMCRS::FacSimcrsServiceContext, 92
  - ~FacSimcrsServiceContext, 93
  - \_pool, 94
  - clean, 94
  - create, 94
  - FacSimcrsServiceContext, 93
  - instance, 93
  - ServicePool\_T, 93
- SIMCRS::FacSupervisor, 94
  - ~FacSupervisor, 95
  - BomFactoryPool\_T, 95

- cleanBomLayer, 96
- cleanFactory, 96
- cleanServiceLayer, 96
- FacSupervisor, 95
- instance, 96
- registerBomFactory, 96
- registerServiceFactory, 96
- ServiceFactoryPool\_T, 95
- SIMCRS::SIMCRS\_Service, 98
  - buildSampleBom, 101
  - buildSampleBookingRequest, 102
  - buildSampleTravelSolutions, 101
  - calculateAvailability, 101
  - calculateSegmentPathList, 100
  - csvDisplay, 103
  - fareQuote, 101
  - initSnapshotAndRMEvents, 100
  - jsonExport, 103
  - optimise, 101
  - parseAndLoad, 100
  - playCancellation, 101
  - sell, 101
  - takeSnapshots, 101
- SIMCRS::SIMCRS\_ServiceContext, 104
  - FacSimcrsServiceContext, 105
  - fromStream, 104
  - toStream, 104
- SIMCRS::ServiceAbstract, 97
  - ~ServiceAbstract, 98
  - fromStream, 98
  - ServiceAbstract, 98
  - toStream, 98
- SIMCRS\_Service
  - SIMCRS::DistributionManager, 88
  - SIMCRS::SIMCRS\_Service, 99
  - SIMCRS::SIMCRS\_ServiceContext, 105
- SIMCRS\_ServicePtr\_T
  - SIMCRS, 85
- STDAIR\_SAMPLE\_DIR
  - simcrs-paths.hpp, 120
  - simcrs-paths.hpp.in, 122
- SYSCONFDIR
  - simcrs-paths.hpp, 119
  - simcrs-paths.hpp.in, 122
- sell
  - SIMCRS::SIMCRS\_Service, 101
- ServiceAbstract
  - SIMCRS::ServiceAbstract, 98
- ServiceAbstract.hpp
  - operator<<, 131
  - operator>>, 131
- ServiceFactoryPool\_T
  - SIMCRS::FacSupervisor, 95
- ServicePool\_T
  - SIMCRS::FacServiceAbstract, 91
  - SIMCRS::FacSimcrsServiceContext, 93
- simcrs-paths.hpp
  - BINDIR, 119
  - DATADIR, 120
  - DATAROOTDIR, 120
  - DOCDIR, 120
  - EXEC\_PREFIX, 119
  - HTMLDIR, 120
  - INCLUDEDIR, 120
  - INFODIR, 120
  - LIBDIR, 119
  - LIBEXECDIR, 119
  - MANDIR, 120
  - PACKAGE, 119
  - PACKAGE\_NAME, 119
  - PACKAGE\_VERSION, 119
  - PDFDIR, 120
  - PREFIXDIR, 119
  - SBINDIR, 119
  - STDAIR\_SAMPLE\_DIR, 120
  - SYSCONFDIR, 119
- simcrs-paths.hpp.in
  - BINDIR, 121
  - DATADIR, 122
  - DATAROOTDIR, 122
  - DOCDIR, 122
  - EXEC\_PREFIX, 121
  - HTMLDIR, 122
  - INCLUDEDIR, 122
  - INFODIR, 122
  - LIBDIR, 121
  - LIBEXECDIR, 121
  - MANDIR, 122
  - PACKAGE, 121
  - PACKAGE\_NAME, 121
  - PACKAGE\_VERSION, 121
  - PDFDIR, 122
  - PREFIXDIR, 121
  - SBINDIR, 122
  - STDAIR\_SAMPLE\_DIR, 122
  - SYSCONFDIR, 122
- simcrs.cpp
  - main, 109
  - operator<<, 108
  - readConfiguration, 108
- simcrs/SIMCRS\_Service.hpp, 143, 144
- simcrs/SIMCRS\_Types.hpp, 145, 146
- simcrs/basic/BasConst.cpp, 105, 106
- simcrs/basic/BasConst\_General.hpp, 106
- simcrs/basic/BasConst\_SIMCRS\_Service.hpp, 106
- simcrs/batches/simcrs.cpp, 107, 109
- simcrs/bom/BomAbstract.cpp, 114
- simcrs/bom/BomAbstract.hpp, 114, 115
- simcrs/command/DistributionManager.cpp, 116
- simcrs/command/DistributionManager.hpp, 117, 118
- simcrs/config/simcrs-paths.hpp, 118, 120
- simcrs/config/simcrs-paths.hpp.in, 121, 122
- simcrs/factory/FacBomAbstract.cpp, 123
- simcrs/factory/FacBomAbstract.hpp, 124
- simcrs/factory/FacServiceAbstract.cpp, 125
- simcrs/factory/FacServiceAbstract.hpp, 125

simcrs/factory/FacSimcrsServiceContext.cpp, [126](#)  
simcrs/factory/FacSimcrsServiceContext.hpp, [127](#)  
simcrs/factory/FacSupervisor.cpp, [128](#)  
simcrs/factory/FacSupervisor.hpp, [129](#)  
simcrs/service/SIMCRS\_Service.cpp, [132](#)  
simcrs/service/SIMCRS\_ServiceContext.cpp, [140](#)  
simcrs/service/SIMCRS\_ServiceContext.hpp, [141](#), [142](#)  
simcrs/service/ServiceAbstract.cpp, [130](#)  
simcrs/service/ServiceAbstract.hpp, [130](#), [131](#)  
stdair, [85](#)

takeSnapshots  
    SIMCRS::SIMCRS\_Service, [101](#)

test/simcrs/CRSTestSuite.cpp, [146](#)

toStream  
    SIMCRS::BomAbstract, [87](#)  
    SIMCRS::ServiceAbstract, [98](#)  
    SIMCRS::SIMCRS\_ServiceContext, [104](#)

toString  
    SIMCRS::BomAbstract, [87](#)