SEvMgr

0.2.0

Generated by Doxygen 1.7.5

Fri Feb 10 2012 10:09:19

Contents

1 SEvMgr Documentation **1**

1.1 Getting Started . . . 1
1.2 SEvMgr at SourceForge . . . 2
1.3 SEvMgr Development . . . 2
1.4 External Libraries . . . 2
1.5 Support SEvMgr . . . 3
1.6 About SEvMgr . . . 3

2 People **3**

2.1 Project Admins . . . 3
2.2 Developers . . . 3
2.3 Retired Developers . . . 3
2.4 Contributors . . . 4
2.5 Distribution Maintainers . . . 4

3 Coding Rules **4**

3.1 Default Naming Rules for Variables . . . 4
3.2 Default Naming Rules for Functions . . . 4
3.3 Default Naming Rules for Classes and Structures . . . 4
3.4 Default Naming Rules for Files . . . 5
3.5 Default Functionality of Classes . . . 5

4 Copyright and License **5**

4.1 GNU LESSER GENERAL PUBLIC LICENSE . . . 5
4.1.1 Version 2.1, February 1999 . . . 5
4.2 Preamble . . . 5
4.3 TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION . . . 7
4.3.1 NO WARRANTY . . . 12
4.3.2 END OF TERMS AND CONDITIONS . . . 12
4.4 How to Apply These Terms to Your New Programs . . . 12

5 Documentation Rules **13**

5.1 General Rules . . . 13

5.2 File Header 14
5.3 Grouping Various Parts 15

6 Main features **15**
6.1 Booking management 15
6.2 Revenue Management notification 15
6.3 Setting simulation break-points 16
6.4 Other features 16

7 Make a Difference **16**

8 Make a new release **16**
8.1 Introduction 17
8.2 Initialisation 17
8.3 Branch creation 17
8.4 Commit and publish the release branch 17
8.5 Update the change-log in the trunk as well 17
8.6 Create distribution packages 18
8.7 Generation the RPM packages 18
8.8 Update distributed change log 18
8.9 Create the binary package, including the documentation 19
8.10 Upload the files to SourceForge 19
8.11 Upload the documentation to SourceForge 19
8.12 Make a new post 19
8.13 Send an email on the announcement mailing-list 20

9 Installation **20**
9.1 Table of Contents 20
9.2 Fedora/RedHat Linux distributions 20
9.3 SEvMgr Requirements 20
9.4 Basic Installation 21
9.5 Compilers and Options 22
9.6 Compiling For Multiple Architectures 23
9.7 Installation Names 23
9.8 Optional Features 24

9.9 Particular systems . . . 25
9.10 Specifying the System Type . . . 26
9.11 Sharing Defaults . . . 26
9.12 Defining Variables . . . 26
9.13 'cmake' Invocation . . . 27

10 Linking with SEvMgr **31**
10.1 Table of Contents . . . 31
10.2 Introduction . . . 32
10.3 Dependencies . . . 32
10.3.1 StdAir . . . 32
10.4 Using the pkg-config command . . . 32
10.5 Using the sevmgr-config script . . . 33
10.6 M4 macro for the GNU Autotools . . . 33
10.7 Using SEvMgr with dynamic linking . . . 34

11 Test Rules **34**
11.1 The Test File . . . 34
11.2 The Reference File . . . 34
11.3 Testing IT++ Library . . . 34

12 Users Guide **35**
12.1 Table of Contents . . . 35
12.2 Introduction . . . 35
12.3 Get Started . . . 35
12.3.1 Get the SEvMgr library . . . 35
12.3.2 Build the SEvMgr project . . . 36
12.3.3 Build and Run the Tests . . . 36
12.3.4 Install the SEvMgr Project (Binaries, Documentation) . . . 36
12.4 Input file of SEvMgr Project . . . 37
12.5 The schedule BOM Tree . . . 38
12.5.1 Build of the schedule BOM tree . . . 38
12.5.2 Display of the schedule BOM tree . . . 39
12.6 Exploring the Predefined BOM Tree . . . 90
12.6.1 Airline Network BOM Tree . . . 90

12.6.2 Airline Schedule BOM Tree . . . 90
12.7 Extending the BOM Tree . . . 91
12.8 The travel solution calculation procedure . . . 91

13 Supported Systems **91**
13.1 Table of Contents . . . 91
13.2 Introduction . . . 91

14 SEvMgr Supported Systems (Previous Releases) **92**
14.1 SEvMgr 3.9.1 . . . 92
14.2 SEvMgr 3.9.0 . . . 92
14.3 SEvMgr 3.8.1 . . . 92

15 Tutorials **92**
15.1 Table of Contents . . . 92
15.2 Preparing the AirSched Project for Development . . . 92
15.3 Your first networkBuilde . . . 92
15.3.1 Summary of the different steps . . . 92
15.3.2 Result of the Batch Program . . . 93
15.4 Network building with an input file . . . 94
15.4.1 How to build a network input file? . . . 94
15.4.2 Building the BOM tree with an input file . . . 95
15.4.3 Result of the Batch Program . . . 95

16 Command-Line Test to Demonstrate How To Use Sevmgr elements **95**

17 Directory Hierarchy **97**
17.1 Directories . . . 97

18 Namespace Index **97**
18.1 Namespace List . . . 97

19 Class Index **97**
19.1 Class Hierarchy . . . 97

20 Class Index **103**
20.1 Class List . . . 103

21 File Index **109**

21.1 File List 109

22 Directory Documentation **110**

22.1 sevmgr/basic/ Directory Reference 110

22.2 sevmgr/batches/ Directory Reference 111

22.3 sevmgr/ui/cmdline/ Directory Reference 111

22.4 sevmgr/command/ Directory Reference 111

22.5 sevmgr/config/ Directory Reference 111

22.6 sevmgr/factory/ Directory Reference 111

22.7 sevmgr/python/ Directory Reference 112

22.8 sevmgr/service/ Directory Reference 112

22.9 test/sevmgr/ Directory Reference 112

22.10 sevmgr/ Directory Reference 112

22.11 test/ Directory Reference 112

22.12 sevmgr/ui/ Directory Reference 113

23 Namespace Documentation **113**

23.1 SEVMGR Namespace Reference 113

23.1.1 Typedef Documentation 114

23.2 stdair Namespace Reference 116

23.3 swift Namespace Reference 116

23.3.1 Detailed Description 116

24 Class Documentation **116**

24.1 std::allocator Class Reference 116

24.1.1 Detailed Description 116

24.2 std::auto_ptr Class Reference 116

24.2.1 Detailed Description 116

24.3 std::bad_alloc Class Reference 116

24.3.1 Detailed Description 117

24.4 std::bad_cast Class Reference 117

24.4.1 Detailed Description 117

24.5 std::bad_exception Class Reference 117

24.5.1 Detailed Description 118

24.6 std::bad_typeid Class Reference . . . 118
24.6.1 Detailed Description . . . 118
24.7 std::basic_fstream Class Reference . . . 118
24.7.1 Detailed Description . . . 118
24.8 std::basic_ifstream Class Reference . . . 118
24.8.1 Detailed Description . . . 119
24.9 std::basic_ios Class Reference . . . 119
24.9.1 Detailed Description . . . 119
24.10 std::basic_iostream Class Reference . . . 119
24.10.1 Detailed Description . . . 119
24.11 std::basic_istream Class Reference . . . 120
24.11.1 Detailed Description . . . 120
24.12 std::basic_istringstream Class Reference . . . 120
24.12.1 Detailed Description . . . 120
24.13 std::basic_ofstream Class Reference . . . 120
24.13.1 Detailed Description . . . 121
24.14 std::basic_ostream Class Reference . . . 121
24.14.1 Detailed Description . . . 121
24.15 std::basic_ostringstream Class Reference . . . 121
24.15.1 Detailed Description . . . 121
24.16 std::basic_string Class Reference . . . 122
24.16.1 Detailed Description . . . 122
24.17 std::basic_stringstream Class Reference . . . 122
24.17.1 Detailed Description . . . 122
24.18 std::bitset Class Reference . . . 122
24.18.1 Detailed Description . . . 123
24.19 CmdAbstract Class Reference . . . 123
24.20 COMMAND Struct Reference . . . 123
24.20.1 Detailed Description . . . 123
24.20.2 Member Data Documentation . . . 123
24.21 std::complex Class Reference . . . 124
24.21.1 Detailed Description . . . 124
24.22 std::list::const_iterator Class Reference . . . 124
24.22.1 Detailed Description . . . 124

24.23std::map::const_iterator Class Reference 124
24.23.1 Detailed Description 124
24.24std::multimap::const_iterator Class Reference 125
24.24.1 Detailed Description 125
24.25std::set::const_iterator Class Reference 125
24.25.1 Detailed Description 125
24.26std::multiset::const_iterator Class Reference 125
24.26.1 Detailed Description 125
24.27std::basic_string::const_iterator Class Reference 125
24.27.1 Detailed Description 125
24.28std::vector::const_iterator Class Reference 125
24.28.1 Detailed Description 126
24.29std::string::const_iterator Class Reference 126
24.29.1 Detailed Description 126
24.30std::wstring::const_iterator Class Reference 126
24.30.1 Detailed Description 126
24.31std::deque::const_iterator Class Reference 126
24.31.1 Detailed Description 126
24.32std::map::const_reverse_iterator Class Reference 126
24.32.1 Detailed Description 126
24.33std::list::const_reverse_iterator Class Reference 127
24.33.1 Detailed Description 127
24.34std::multimap::const_reverse_iterator Class Reference 127
24.34.1 Detailed Description 127
24.35std::set::const_reverse_iterator Class Reference 127
24.35.1 Detailed Description 127
24.36std::multiset::const_reverse_iterator Class Reference 127
24.36.1 Detailed Description 127
24.37std::basic_string::const_reverse_iterator Class Reference 127
24.37.1 Detailed Description 128
24.38std::vector::const_reverse_iterator Class Reference 128
24.38.1 Detailed Description 128
24.39std::string::const_reverse_iterator Class Reference 128
24.39.1 Detailed Description 128

24.40 std::wstring::const_reverse_iterator Class Reference . . . 128
24.40.1 Detailed Description . . . 128
24.41 std::deque::const_reverse_iterator Class Reference . . . 128
24.41.1 Detailed Description . . . 128
24.42 std::deque Class Reference . . . 129
24.42.1 Detailed Description . . . 129
24.43 std::domain_error Class Reference . . . 129
24.43.1 Detailed Description . . . 129
24.44 SEVMGR::EventQueueManager Class Reference . . . 129
24.44.1 Detailed Description . . . 130
24.44.2 Friends And Related Function Documentation . . . 130
24.45 std::exception Class Reference . . . 130
24.45.1 Detailed Description . . . 130
24.46 FacServiceAbstract Class Reference . . . 131
24.47 SEVMGR::FacSEVMGRServiceContext Class Reference . . . 131
24.47.1 Detailed Description . . . 131
24.47.2 Constructor & Destructor Documentation . . . 132
24.47.3 Member Function Documentation . . . 132
24.48 std::ios_base::failure Class Reference . . . 133
24.48.1 Detailed Description . . . 133
24.49 std::fstream Class Reference . . . 133
24.49.1 Detailed Description . . . 133
24.50 std::ifstream Class Reference . . . 133
24.50.1 Detailed Description . . . 134
24.51 std::invalid_argument Class Reference . . . 134
24.51.1 Detailed Description . . . 134
24.52 std::ios Class Reference . . . 134
24.52.1 Detailed Description . . . 134
24.53 std::ios_base Class Reference . . . 135
24.53.1 Detailed Description . . . 135
24.54 std::istream Class Reference . . . 135
24.54.1 Detailed Description . . . 135
24.55 std::istringstream Class Reference . . . 135
24.55.1 Detailed Description . . . 136

24.56std::list::iterator Class Reference . . . 136
24.56.1 Detailed Description . . . 136
24.57std::deque::iterator Class Reference . . . 136
24.57.1 Detailed Description . . . 136
24.58std::multimap::iterator Class Reference . . . 136
24.58.1 Detailed Description . . . 136
24.59std::set::iterator Class Reference . . . 137
24.59.1 Detailed Description . . . 137
24.60std::multiset::iterator Class Reference . . . 137
24.60.1 Detailed Description . . . 137
24.61std::basic_string::iterator Class Reference . . . 137
24.61.1 Detailed Description . . . 137
24.62std::vector::iterator Class Reference . . . 137
24.62.1 Detailed Description . . . 137
24.63std::string::iterator Class Reference . . . 137
24.63.1 Detailed Description . . . 138
24.64std::map::iterator Class Reference . . . 138
24.64.1 Detailed Description . . . 138
24.65std::wstring::iterator Class Reference . . . 138
24.65.1 Detailed Description . . . 138
24.66std::length_error Class Reference . . . 138
24.66.1 Detailed Description . . . 138
24.67std::list Class Reference . . . 139
24.67.1 Detailed Description . . . 139
24.68std::logic_error Class Reference . . . 139
24.68.1 Detailed Description . . . 139
24.69std::map Class Reference . . . 139
24.69.1 Detailed Description . . . 140
24.70std::multimap Class Reference . . . 140
24.70.1 Detailed Description . . . 140
24.71std::multiset Class Reference . . . 140
24.71.1 Detailed Description . . . 141
24.72std::ofstream Class Reference . . . 141
24.72.1 Detailed Description . . . 141

24.73std::ostream Class Reference . . . 141
24.73.1 Detailed Description . . . 142
24.74std::ostringstream Class Reference . . . 142
24.74.1 Detailed Description . . . 142
24.75std::out_of_range Class Reference . . . 142
24.75.1 Detailed Description . . . 142
24.76std::overflow_error Class Reference . . . 143
24.76.1 Detailed Description . . . 143
24.77std::priority_queue Class Reference . . . 143
24.77.1 Detailed Description . . . 143
24.78SEVMGR::PYEventQueueManager Struct Reference . . . 143
24.78.1 Constructor & Destructor Documentation . . . 144
24.78.2 Member Function Documentation . . . 144
24.79std::queue Class Reference . . . 144
24.79.1 Detailed Description . . . 145
24.80std::range_error Class Reference . . . 145
24.80.1 Detailed Description . . . 145
24.81std::vector::reverse_iterator Class Reference . . . 145
24.81.1 Detailed Description . . . 145
24.82std::list::reverse_iterator Class Reference . . . 145
24.82.1 Detailed Description . . . 145
24.83std::map::reverse_iterator Class Reference . . . 146
24.83.1 Detailed Description . . . 146
24.84std::deque::reverse_iterator Class Reference . . . 146
24.84.1 Detailed Description . . . 146
24.85std::wstring::reverse_iterator Class Reference . . . 146
24.85.1 Detailed Description . . . 146
24.86std::basic_string::reverse_iterator Class Reference . . . 146
24.86.1 Detailed Description . . . 146
24.87std::multimap::reverse_iterator Class Reference . . . 146
24.87.1 Detailed Description . . . 147
24.88std::set::reverse_iterator Class Reference . . . 147
24.88.1 Detailed Description . . . 147
24.89std::multiset::reverse_iterator Class Reference . . . 147

24.89.1 Detailed Description . . . 147
24.90 std::string::reverse_iterator Class Reference . . . 147
24.90.1 Detailed Description . . . 147
24.91 RootException Class Reference . . . 147
24.92 std::runtime_error Class Reference . . . 148
24.92.1 Detailed Description . . . 148
24.93 ServiceAbstract Class Reference . . . 148
24.94 std::set Class Reference . . . 148
24.94.1 Detailed Description . . . 149
24.95 SEVMGR::SEVMGR_Service Class Reference . . . 149
24.95.1 Detailed Description . . . 149
24.95.2 Constructor & Destructor Documentation . . . 150
24.95.3 Member Function Documentation . . . 151
24.96 SEVMGR::SEVMGR_ServiceContext Class Reference . . . 152
24.96.1 Detailed Description . . . 152
24.96.2 Friends And Related Function Documentation . . . 152
24.97 SEVMGR::SEvMgrException Class Reference . . . 153
24.97.1 Detailed Description . . . 153
24.97.2 Constructor & Destructor Documentation . . . 153
24.98 swift::SKeymap Class Reference . . . 154
24.98.1 Detailed Description . . . 154
24.98.2 Constructor & Destructor Documentation . . . 154
24.98.3 Member Function Documentation . . . 155
24.98.4 Friends And Related Function Documentation . . . 156
24.99 swift::SReadline Class Reference . . . 156
24.99.1 Detailed Description . . . 157
24.99.2 Constructor & Destructor Documentation . . . 157
24.99.3 Member Function Documentation . . . 158
24.100 std::stack Class Reference . . . 161
24.100.1 Detailed Description . . . 162
24.101 std::string Class Reference . . . 162
24.101.1 Detailed Description . . . 162
24.102 std::stringstream Class Reference . . . 162
24.102.1 Detailed Description . . . 163

24.103 std::underflow_error Class Reference . . . 163
24.103.1 Detailed Description . . . 163
24.104 std::valarray Class Reference . . . 163
24.104.1 Detailed Description . . . 163
24.105 std::vector Class Reference . . . 163
24.105.1 Detailed Description . . . 164
24.106 std::wfstream Class Reference . . . 164
24.106.1 Detailed Description . . . 164
24.107 std::wifstream Class Reference . . . 164
24.107.1 Detailed Description . . . 164
24.108 std::wios Class Reference . . . 165
24.108.1 Detailed Description . . . 165
24.109 std::wistream Class Reference . . . 165
24.109.1 Detailed Description . . . 165
24.110 std::wistringstream Class Reference . . . 165
24.110.1 Detailed Description . . . 166
24.111 std::wofstream Class Reference . . . 166
24.111.1 Detailed Description . . . 166
24.112 std::wostream Class Reference . . . 166
24.112.1 Detailed Description . . . 166
24.113 std::wostringstream Class Reference . . . 167
24.113.1 Detailed Description . . . 167
24.114 std::wstring Class Reference . . . 167
24.114.1 Detailed Description . . . 168
24.115 std::wstringstream Class Reference . . . 168
24.115.1 Detailed Description . . . 168

25 File Documentation **168**

25.1 doc/local/authors.doc File Reference . . . 168
25.2 doc/local/codingrules.doc File Reference . . . 168
25.3 doc/local/copyright.doc File Reference . . . 168
25.4 doc/local/documentation.doc File Reference . . . 168
25.5 doc/local/features.doc File Reference . . . 168
25.6 doc/local/help_wanted.doc File Reference . . . 168

25.7 doc/local/howto_release.doc File Reference 168
25.8 doc/local/index.doc File Reference 168
25.9 doc/local/installation.doc File Reference 168
25.10 doc/local/linking.doc File Reference 169
25.11 doc/local/test.doc File Reference 169
25.12 doc/local/users_guide.doc File Reference 169
25.13 doc/local/verification.doc File Reference 169
25.14 doc/tutorial/tutorial.doc File Reference 169
25.15 sevmgr/basic/BasConst.cpp File Reference 169
25.16 BasConst.cpp 169
25.17 sevmgr/basic/BasConst_EventQueueManager.hpp File Reference . . . 169
25.18 BasConst_EventQueueManager.hpp 169
25.19 sevmgr/basic/BasConst_SEVMGR_Service.hpp File Reference 170
25.20 BasConst_SEVMGR_Service.hpp 170
25.21 sevmgr/basic/BasParserTypes.hpp File Reference 170
25.22 BasParserTypes.hpp 171
25.23 sevmgr/batches/sevmgr_demo.cpp File Reference 172
25.23.1 Function Documentation 173
25.23.2 Variable Documentation 173
25.24 sevmgr_demo.cpp 174
25.25 sevmgr/command/EventQueueManager.cpp File Reference 176
25.26 EventQueueManager.cpp 176
25.27 sevmgr/command/EventQueueManager.hpp File Reference 176
25.28 EventQueueManager.hpp 177
25.29 sevmgr/config/sevmgr-paths.hpp File Reference 177
25.29.1 Define Documentation 178
25.30 sevmgr-paths.hpp 179
25.31 sevmgr/config/sevmgr-paths.hpp.in File Reference 180
25.31.1 Define Documentation 180
25.32 sevmgr-paths.hpp.in 182
25.33 sevmgr/factory/FacSEVMGRServiceContext.cpp File Reference 182
25.34 FacSEVMGRServiceContext.cpp 182
25.35 sevmgr/factory/FacSEVMGRServiceContext.hpp File Reference 183
25.36 FacSEVMGRServiceContext.hpp 183

25.37 sevmgr/python/pysevmgr.cpp File Reference . . . 184
25.37.1 Function Documentation . . . 184
25.38 pysevmgr.cpp . . . 185
25.39 sevmgr/service/SEVMGR_Service.cpp File Reference . . . 187
25.40 SEVMGR_Service.cpp . . . 187
25.41 sevmgr/service/SEVMGR_ServiceContext.cpp File Reference . . . 191
25.42 SEVMGR_ServiceContext.cpp . . . 191
25.43 sevmgr/service/SEVMGR_ServiceContext.hpp File Reference . . . 192
25.44 SEVMGR_ServiceContext.hpp . . . 192
25.45 sevmgr/SEVMGR_Exceptions.hpp File Reference . . . 193
25.46 SEVMGR_Exceptions.hpp . . . 194
25.47 sevmgr/SEVMGR_Service.hpp File Reference . . . 194
25.48 SEVMGR_Service.hpp . . . 195
25.49 sevmgr/SEVMGR_Types.hpp File Reference . . . 196
25.50 SEVMGR_Types.hpp . . . 196
25.51 sevmgr/ui/cmdline/readline_autocomp.hpp File Reference . . . 196
25.51.1 Typedef Documentation . . . 197
25.51.2 Function Documentation . . . 198
25.51.3 Variable Documentation . . . 200
25.52 readline_autocomp.hpp . . . 201
25.53 sevmgr/ui/cmdline/sevmgr.cpp File Reference . . . 205
25.54 sevmgr.cpp . . . 205
25.55 sevmgr/ui/cmdline/SReadline.hpp File Reference . . . 216
25.55.1 Detailed Description . . . 217
25.56 SReadline.hpp . . . 217
25.57 test/sevmgr/EventQueueManagementTestSuite.cpp File Reference . . . 223
25.58 EventQueueManagementTestSuite.cpp . . . 223

1 SEvMgr Documentation

1.1 Getting Started

- Main features
- Installation

- Linking with SEvMgr
- Users Guide
- Tutorials
- Copyright and License
- Make a Difference
- Make a new release
- People

1.2 SEvMgr at SourceForge

- Project page
- Download SEvMgr
- Open a ticket for a bug or feature
- Mailing lists
- Forums
  - Discuss about Development issues
  - Ask for Help
  - Discuss SEvMgr

1.3 SEvMgr Development

- Git Repository
- Coding Rules
- Documentation Rules
- Test Rules

1.4 External Libraries

- Boost (C++ STL extensions)
- Python
- MySQL client
- SOCI (C++ DB API)

1.5 Support SEvMgr

1.6 About SEvMgr

SEvMgr is a C++ library of discrete event queue management classes and functions, exclusively targeting simulation purposes. N

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

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

2 People

2.1 Project Admins

- Gabrielle Sabatier <gsabatier@users.sourceforge.net> (N)
- Denis Arnaud <denis_arnaud@users.sourceforge.net> (N)

2.2 Developers

- Anh Quan Nguyen <quannaus@users.sourceforge.net> (N)
- Denis Arnaud <denis_arnaud@users.sourceforge.net> (N)

2.3 Retired Developers

- Mehdi Ayouni <mehdi.ayouni@gmail.com>
- Patrick Grandjean <pgrandjean@users.sourceforge.net> (N)

2.4 Contributors

- Emmanuel Bastien <ebastien@users.sourceforge.net> (N)

2.5 Distribution Maintainers

- Fedora/RedHat: Denis Arnaud <denis_arnaud@users.sourceforge.net> (N)
- Debian: Emmanuel Bastien <ebastien@users.sourceforge.net> (N)

Note

(N) - Amadeus employees.

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.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if

you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include

the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the -

Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

4.3.1 NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

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 SEvMgr 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 SEvMgr 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.
```

```
 *
 * ----------------------------------------------------------------------
 *
 * SEvMgr - C++ Airline Inventory Management Library
 *
 * Copyright (C) 2009-2010  (\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 SEvMgr is given below sorted in different categories. Many more features and functions exist and for these we refer to the reference documentation.

6.1 Booking management

- Booking and cancellation requests

6.2 Revenue Management notification

- Forecast and Optimisation notification requests

6.3 Setting simulation break-points

- Simulation break-points

6.4 Other features

- CSV input file parsing
- Memory handling

7 Make a Difference

Do not ask what SEvMgr can do for you. Ask what you can do for SEvMgr.

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

- Start using SEvMgr
- Tell your friends about SEvMgr 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 SEvMgr discussion forums on SourceForge. - If you know the answer to a question, help others to overcome their SEvMgr 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 SEvMgr to new platforms. If you manage to compile SEvMgr on a new platform, then tell us how you did it.
- Send us your code. If you have a good SEvMgr compatible code, which you can release under the LGPLv2.1, and you think it should be included in SEvMgr, then send it to us.
- Become an SEvMgr developer. Send us an e-mail and tell what you can do for SEvMgr.

8 Make a new release

8.1 Introduction

This document describes briefly the recommended procedure of releasing a new version of SEvMgr 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://sevmgr.git.sourceforge.net/gitroot/sevmgr/sevmgr sevmgrgit
cd sevmgrgit
git checkout trunk
```

8.3 Branch creation

Create the branch, on your local clone, corresponding to the new release (say, 0.5.0):

```
cd ~/dev/sim/sevmgrgit
git checkout trunk
git checkout -b 0.5.0
```

Update the version in the various build system files, replacing 99.99.99 by the correct version number:

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

Update the version and add a change-log in the ChangeLog and in the RPM specification files:

```
vi ChangeLog
vi sevmgr.spec
```

8.4 Commit and publish the release branch

Commit the new release:

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

8.5 Update the change-log in the trunk as well

Update the change-log in the ChangeLog and RPM specification files:

```
cd ~/dev/sim/sevmgrgit
git checkout trunk
vi ChangeLog
vi sevmgr.spec
```

Commit the change-logs and publish the trunk (main development branch):

```
git commit -m "[Doc] Integrated the change-log of the release 0.5.0."
git push
```

8.6 Create distribution packages

Create the distribution packages using the following command:

```
cd ~/dev/sim/sevmgrgit
git checkout 0.5.0
rm -rf build && mkdir -p build
cd build
cmake -DCMAKE_INSTALL_PREFIX=/home/user/dev/deliveries/sevmgr-0.5.0 \
 -DWITH_STDAIR_PREFIX=/home/user/dev/deliveries/stdair-stable \
 -DCMAKE_BUILD_TYPE:STRING=Debug -DINSTALL_DOC:BOOL=ON ..
make check && make dist
```

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

8.7 Generation the RPM packages

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

```
cd ~/dev/sim/sevmgrgit
git checkout 0.5.0
rm -rf build && mkdir -p build
cd build
cmake -DCMAKE_INSTALL_PREFIX=/home/user/dev/deliveries/sevmgr-0.5.0 \
 -DWITH_STDAIR_PREFIX=/home/user/dev/deliveries/stdair-stable \
 -DCMAKE_BUILD_TYPE:STRING=Debug -DINSTALL_DOC:BOOL=ON ..
make dist
```

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

```
cp sevmgr.spec ~/dev/packages/SPECS \
  && cp sevmgr-0.5.0.tar.bz2 ~/dev/packages/SOURCES
cd ~/dev/packages/SPECS
rpmbuild -ba sevmgr.spec
rpmlint -i ../SPECS/sevmgr.spec ../SRPMS/sevmgr-0.5.0-1.fc15.src.rpm \
  ../RPMS/noarch/sevmgr-* ../RPMS/i686/sevmgr-*
```

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 SEvMgr'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:

```
make package
```

The output binary package will be named, for instance, `sevmgr-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 Upload the documentation to SourceForge

In order to update the Web site files, either:

- synchronise them with rsync and SSH:

  ```
  cd ~/dev/sim/sevmgrgit
  git checkout 0.5.0
  rsync -aiv doc/html/ doc/latex/refman.pdf joe,sevmgr@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.

8.12 Make a new post

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

8.13 Send an email on the announcement mailing-list

Finally, you should send an announcement to sevmgr-announce@lists.-sourceforge.net (see https://lists.sourceforge.net/lists/listinfo/sevmgr-announce for the archives)

9 Installation

9.1 Table of Contents

- Fedora/RedHat Linux distributions
- SEvMgr 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 sevmgr-devel sevmgr-doc
```

RPM packages can also be available on the SourceForge download site.

9.3 SEvMgr Requirements

SEvMgr 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 SEvMgr.

9.4 Basic Installation

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

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

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

If you need to do unusual things to compile the package, please try to figure out how 'configure' could check whether to do them, and mail diffs or instructions to the address given in the 'README' so they can be considered for the next release. If you are using

the cache, and at some point 'config.cache' contains results you don't want to keep, you may remove or edit it.

The file 'CMakeLists.txt' is used to create the 'Makefile' files.

The simplest way to compile this package is:

1. 'cd' to the directory containing the package's source code and type './cmake .. ' to configure the package for your system. Running 'cmake' is generally fast. While running, it prints some messages telling which features it is checking for.
2. Type 'make' to compile the package.
3. Optionally, type 'make check'to run any self-tests that come with the package, generally using the just-built uninstalled binaries.
4. Type 'make install' to install the programs and any data files and documentation. When installing into a prefix owned by root, it is recommended that the package be configured and built as a regular user, and only the 'make install' phase executed with root privileges.
5. You can remove the program binaries and object files from the source code directory by typing 'make clean'. To also remove the files that 'configure' created (so you can compile the package for a different kind of computer), type 'make distclean'. There is also a 'make maintainer-clean' target, but that is intended mainly for the package's developers. If you use it, you may have to get all sorts of other programs in order to regenerate files that came with the distribution.
6. Often, you can also type 'make uninstall' to remove the installed files again. In practice, not all packages have tested that uninstallation works correctly, even though it is required by the GNU Coding Standards.

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 ouput like this:

```
export LIBSUFFIX_4_CMAKE="-DLIB_SUFFIX=64"
export INSTALL_BASEDIR=/home/user/dev/deliveries
cmake -DCMAKE_INSTALL_PREFIX=${INSTALL_BASEDIR}/sevmgr-0.5.0 \
 -DWITH_STDAIR_PREFIX=${INSTALL_BASEDIR}/stdair-stable \
 -DWITH_AIRRAC_PREFIX=${INSTALL_BASEDIR}/airrac-stable \
 -DWITH_RMOL_PREFIX=${INSTALL_BASEDIR}/rmol-stable \
 -DCMAKE_BUILD_TYPE:STRING=Debug -DINSTALL_DOC:BOOL=ON ${LIBSUFFIX_4_CMAKE} ..
-- The C compiler identification is GNU
-- The CXX compiler identification is GNU
-- Check for working C compiler: /usr/lib64/ccache/gcc
-- Check for working C compiler: /usr/lib64/ccache/gcc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working CXX compiler: /usr/lib64/ccache/c++
-- Check for working CXX compiler: /usr/lib64/ccache/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Requires Git without specifying any version
-- Current Git revision name: 0ee8dcc3e3dd1d1d442c4054fbfa4cacc1182e6a trunk
-- Requires Boost-1.41
-- Boost version: 1.46.0
-- Found the following Boost libraries:
--   regex
--   program_options
--   date_time
--   iostreams
--   serialization
--   filesystem
--   unit_test_framework
--   python
-- Found Boost version: 1.46.0
-- Found BoostWrapper: /usr/include (Required is at least version "1.41")
-- Requires Readline without specifying any version
-- Found Readline: /usr/include
-- Found Readline version: 6.2
-- Requires MySQL without specifying any version
-- Using mysql-config: /usr/bin/mysql_config
-- Found MySQL: /usr/lib64/mysql/libmysqlclient.so
-- Found MySQL version: 5.5.14
-- Requires SOCI-3.0
-- Using soci-config: /usr/bin/soci-config
-- SOCI headers are buried
-- Found SOCI: /usr/lib64/libsoci_core.so (Required is at least version "3.0")
-- Found SOCIMySQL: /usr/lib64/libsoci_mysql.so (Required is at least version "3.0")
-- Found SOCI with MySQL back-end support version: 3.0.0
-- Requires StdAir-0.37
-- Found StdAir version: 0.38.0
```

```
-- Requires Doxygen without specifying any version
-- Found Doxygen: /usr/bin/doxygen
-- Found DoxygenWrapper: /usr/bin/doxygen
-- Found Doxygen version: 1.7.4
-- Had to set the linker language for 'sevmgrlib' to CXX
-- Test 'InventoryTestSuite' to be built with 'InventoryTestSuite.cpp'
--
-- ============================================================
-- ----------------------------------
-- ---     Project Information    ---
-- ----------------------------------
-- PROJECT_NAME ................... : sevmgr
-- PACKAGE_PRETTY_NAME ............ : SEvMgr
-- PACKAGE ........................ : sevmgr
-- PACKAGE_NAME ................... : SEVMGR
-- PACKAGE_BRIEF .................. : C++ Simulation-Oriented Discrete Event Management Library
-- PACKAGE_VERSION ................ : 0.5.0
-- GENERIC_LIB_VERSION ............ : 0.5.0
-- GENERIC_LIB_SOVERSION .......... : 0.5
--
-- ----------------------------------
-- ---     Build Configuration    ---
-- ----------------------------------
-- Modules to build ............... : airrac;rmol;sevmgr
-- Libraries to build/install ..... : airraclib;rmollib;sevmgrlib
-- Binaries to build/install ...... : airrac;rmol;sevmgr_parseInventory;sevmgr
-- Modules to test ................ : sevmgr
-- Binaries to test ............... : InventoryTestSuitetst
--
-- * Module ....................... : sevmgr
--   + Layers to build ............ : .;basic;bom;factory;command;service
--   + Dependencies on other layers : airraclib;rmollib
--   + Libraries to build/install . : sevmgrlib
--   + Executables to build/install : sevmgr_parseInventory;sevmgr
--   + Tests to perform ........... : InventoryTestSuitetst
--
-- BUILD_SHARED_LIBS .............. : ON
-- CMAKE_BUILD_TYPE ............... : Debug
--  * CMAKE_C_FLAGS ............... :
--  * CMAKE_CXX_FLAGS ............. : -Wall -Werror
--  * BUILD_FLAGS ................. :
--  * COMPILE_FLAGS ............... :
-- CMAKE_MODULE_PATH .............. : /home/dan/dev/sim/sevmgr/sevmgrgithub/config/
-- CMAKE_INSTALL_PREFIX ........... : /home/dan/dev/deliveries/sevmgr-0.5.0
--
-- * Doxygen:
--   - DOXYGEN_VERSION ............. : 1.7.4
--   - DOXYGEN_EXECUTABLE .......... : /usr/bin/doxygen
--   - DOXYGEN_DOT_EXECUTABLE ...... : /usr/bin/dot
--   - DOXYGEN_DOT_PATH ............ : /usr/bin
--
-- ----------------------------------
-- --- Installation Configuration ---
-- ----------------------------------
-- INSTALL_LIB_DIR ................ : /home/dan/dev/deliveries/sevmgr-0.5.0/lib64
-- INSTALL_BIN_DIR ................ : /home/dan/dev/deliveries/sevmgr-0.5.0/bin
-- INSTALL_INCLUDE_DIR ............ : /home/dan/dev/deliveries/sevmgr-0.5.0/include
-- INSTALL_DATA_DIR ............... : /home/dan/dev/deliveries/sevmgr-0.5.0/share
-- INSTALL_SAMPLE_DIR ............. : /home/dan/dev/deliveries/sevmgr-0.5.0/share/sevmgr/samples
-- INSTALL_DOC .................... : ON
--
-- ----------------------------------
```

```
-- ---   Packaging Configuration  ---
-- ----------------------------------
-- CPACK_PACKAGE_CONTACT ......... : Denis Arnaud <denis_arnaud - at - users dot sourceforge dot n
-- CPACK_PACKAGE_VENDOR ........... : Denis Arnaud
-- CPACK_PACKAGE_VERSION ......... : 0.5.0
-- CPACK_PACKAGE_DESCRIPTION_FILE . : /home/dan/dev/sim/sevmgr/sevmgrgithub/README
-- CPACK_RESOURCE_FILE_LICENSE .... : /home/dan/dev/sim/sevmgr/sevmgrgithub/COPYING
-- CPACK_GENERATOR ................ : TBZ2
-- CPACK_DEBIAN_PACKAGE_DEPENDS ... :
-- CPACK_SOURCE_GENERATOR ......... : TBZ2;TGZ
-- CPACK_SOURCE_PACKAGE_FILE_NAME . : sevmgr-0.5.0
--
-- ----------------------------------
-- ---     External libraries    ---
-- ----------------------------------
--
-- * Boost:
--   - Boost_VERSION .............. : 104600
--   - Boost_LIB_VERSION .......... : 1_46
--   - Boost_HUMAN_VERSION ........ : 1.46.0
--   - Boost_INCLUDE_DIRS ......... : /usr/include
--   - Boost required components .. : regex;program_options;date_time;iostreams;serialization;files
--   - Boost required libraries ... : optimized;/usr/lib64/libboost_regex-mt.so;debug;/usr/lib64/li
--
-- * Readline:
--   - READLINE_VERSION ........... : 6.2
--   - READLINE_INCLUDE_DIR ....... : /usr/include
--   - READLINE_LIBRARY ........... : /usr/lib64/libreadline.so
--
-- * MySQL:
--   - MYSQL_VERSION .............. : 5.5.14
--   - MYSQL_INCLUDE_DIR .......... : /usr/include/mysql
--   - MYSQL_LIBRARIES ............ : /usr/lib64/mysql/libmysqlclient.so
--
-- * SOCI:
--   - SOCI_VERSION ............... : 3.0.0
--   - SOCI_INCLUDE_DIR ........... : /usr/include/soci
--   - SOCIMYSQL_INCLUDE_DIR ...... : /usr/include/soci
--   - SOCI_LIBRARIES ............. : /usr/lib64/libsoci_core.so
--   - SOCIMYSQL_LIBRARIES ........ : /usr/lib64/libsoci_mysql.so
--
-- * StdAir:
--   - STDAIR_VERSION ............. : 0.38.0
--   - STDAIR_BINARY_DIRS ......... : /home/dan/dev/deliveries/stdair-0.38.0/bin
--   - STDAIR_EXECUTABLES ......... : stdair
--   - STDAIR_LIBRARY_DIRS ........ : /home/dan/dev/deliveries/stdair-0.38.0/lib64
--   - STDAIR_LIBRARIES ........... : stdairlib;stdairuicllib
--   - STDAIR_INCLUDE_DIRS ........ : /home/dan/dev/deliveries/stdair-0.38.0/include
--   - STDAIR_SAMPLE_DIR .......... : /home/dan/dev/deliveries/stdair-0.38.0/share/stdair/samples
--
-- Change a value with: cmake -D<Variable>=<Value>
-- ================================================================
--
-- Configuring done
-- Generating done
-- Build files have been written to: /home/dan/dev/sim/sevmgr/sevmgrgithub/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_sevmgr
[  0%] Built target hdr_cfg_airrac
[ 13%] Built target hdr_cfg_rmol
[ 98%] Built target sevmgrlib
[100%] Built target InventoryTestSuitetst
Scanning dependencies of target check_sevmgrtst
Test project /home/dan/dev/sim/sevmgr/sevmgrgithub/build/test/sevmgr
    Start 1: InventoryTestSuitetst
1/1 Test #1: InventoryTestSuitetst .............   Passed    0.08 sec

100% tests passed, 0 tests failed out of 1

Total Test time (real) =   0.35 sec
[100%] Built target check_sevmgrtst
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/sevmgrgit
rm -rf build && mkdir build
cd build
```

to remove everything.

10 Linking with SEvMgr

10.1 Table of Contents

- Introduction
- Dependencies
- Using the pkg-config command

- Using the sevmgr-config script
- M4 macro for the GNU Autotools
- Using SEvMgr with dynamic linking

10.2 Introduction

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

10.3 Dependencies

The SEvMgr 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').

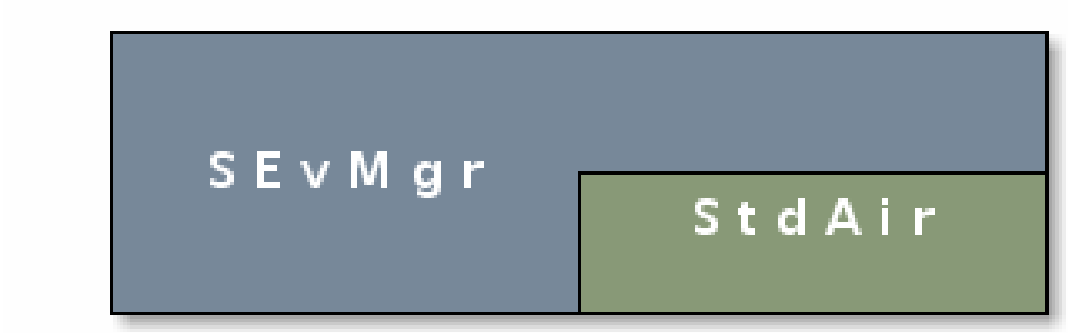


Figure 1: SEvMgr Dependencies

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 SEvMgr based program 'my_prog.cpp', you should use the following command:

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

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

10.5 Using the sevmgr-config script

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

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

```
sevmgr-config --help
```

If the 'sevmgr-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 SEvMgr, namely 'sevmgr.m4', which can be found in, e.g., '/usr/share/aclocal'. When used by a 'configure' script, thanks to he 'AM_PATH_SEvMgr' macro (specified in the M4 macro file), the following Makefile variables are then defined:

- 'SEvMgr_VERSION' (e.g., defined to 0.23.0)
- 'SEvMgr_CFLAGS' (e.g., defined to '-I${prefix}/include')
- 'SEvMgr_LIBS' (e.g., defined to '-L${prefix}/lib -lsevmgr')

10.7 Using SEvMgr with dynamic linking

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

11 Test Rules

This section describes rules how the functionality of the IT++ library should be verified. In the 'tests' subdirectory test files are provided. All functionality should be tested using these test files.

11.1 The Test File

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

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

The test file should be placed in the 'tests' subdirectory and should have a name ending with '_test.cpp'.

11.2 The Reference File

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

11.3 Testing IT++ Library

One can compile and execute all test programs from 'tests' subdirectory by typing

```
% make check
```

after successful compilation of the IT++ library.

12 Users Guide

12.1 Table of Contents

- Introduction
- Get Started
  - Get the SEvMgr library
  - Build the SEvMgr project
  - Build and Run the Tests
  - Install the SEvMgr Project (Binaries, Documentation)
- Input file of SEvMgr 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 `SEvMgr` library contains classes for airline business management. This document does not cover all the aspects of the `SEvMgr` library. It does however explain the most important things you need to know in order to start using `SEvMgr`.

12.3 Get Started

12.3.1 Get the SEvMgr library

Clone locally the full `Git project`:

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

12.3.2 Build the SEvMgr project

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

```
cd ~/dev/sim/sevmgrgit
rm -rf build && mkdir -p build
cd build
cmake -DCMAKE_INSTALL_PREFIX=~/dev/deliveries/sevmgr-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 SEvMgr project, the following commands run the tests:

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

As a result, you should obtain a similar report:

```
[  0%] Built target hdr_cfg_sevmgr
[ 96%] Built target sevmgrlib
[100%] Built target AirlineScheduleTestSuitetst
Scanning dependencies of target check_sevmgrtst
Test project /home/dan/dev/sim/sevmgr/sevmgrgithub/build/test/sevmgr
    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_sevmgrtst
Scanning dependencies of target check
[100%] Built target check
```

12.3.4 Install the SEvMgr Project (Binaries, Documentation)

After the step Build the SEvMgr project, to install the library and its header files, type:

```
cd ~/dev/sim/sevmgrgit
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/sevmgr-0.5.0
```

To generate the SEvMgr project documentation, the commands are:

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

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

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

12.4 Input file of SEvMgr 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 `SEVMGR::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 bahave 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`, `JCDI`, `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`, `JCDI`, `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 `SEVMGR::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 SEvMgr 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 `SEVMGR::SEVMGR_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 `SEVMGR::SEVMGR_Service::buildSampleBom()` method:

```
    void buildSampleBom();
```

or can be constructed using the schedule input file described above thanks to the SEVMGR::SEVMGR_Service::parseAndLoad (const stdair::Filename_T&) method:

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::sevmgr.cpp program:

When the default BOM tree is used (-b/--builtin option of the main program sevmgr.cpp), the schedule BOM tree display (for now, corresponding to schedule01.csv parsed by SEVMGR::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, 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, 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, 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, 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, 300, 0,
SQ11 2010-Jan-26, SIN-BKK 2010-Jan-26, Y, 2, 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, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-27, SIN-BKK 2010-Jan-27, Y, 1, 0, 0, 0, 300, 0,
SQ11 2010-Jan-27, SIN-BKK 2010-Jan-27, Y, 2, 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-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, 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, 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, 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, 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,
SQ11 2010-Feb-05, SIN-BKK 2010-Feb-05, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-05, SIN-BKK 2010-Feb-05, Y, 2, 0, 0, 0, 0, 300, 0,
****************************************
****************************************
Subclasses:
-----------
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
      GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQ11 2010-Feb-05, SIN-BKK 2010-Feb-05, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
SQ11 2010-Feb-05, SIN-BKK 2010-Feb-05, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
****************************************
****************************************
FlightDate: SQ11, 2010-Feb-06
****************************************
****************************************
Leg-Dates:
----------
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
      Elapsed, Distance, Capacity,
SQ11 2010-Feb-06, SIN-BKK, 2010-Feb-06, 08:20:00, 2010-Feb-06, 11:00:00, 07:40:
      00, 0, -05:00:00, 6300, 0,
****************************************
****************************************
LegCabins:
----------
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
      CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
```

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

```
****************************************
FlightDate: SQ11, 2010-Feb-10
****************************************
****************************************
Leg-Dates:
----------
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
      Elapsed, Distance, Capacity,
SQ11 2010-Feb-10, SIN-BKK, 2010-Feb-10, 08:20:00, 2010-Feb-10, 11:00:00, 07:40:
      00, 0, -05:00:00, 6300, 0,
****************************************
****************************************
LegCabins:
----------
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
      CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ11 2010-Feb-10, SIN-BKK 2010-Feb-10, 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-10, SIN-BKK 2010-Feb-10, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-10, SIN-BKK 2010-Feb-10, Y, 2, 0, 0, 0, 0, 300, 0,
****************************************
****************************************
Subclasses:
-----------
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
      GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQ11 2010-Feb-10, SIN-BKK 2010-Feb-10, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
SQ11 2010-Feb-10, SIN-BKK 2010-Feb-10, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
****************************************
****************************************
FlightDate: SQ11, 2010-Feb-11
****************************************
****************************************
Leg-Dates:
----------
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
      Elapsed, Distance, Capacity,
SQ11 2010-Feb-11, SIN-BKK, 2010-Feb-11, 08:20:00, 2010-Feb-11, 11:00:00, 07:40:
      00, 0, -05:00:00, 6300, 0,
****************************************
****************************************
LegCabins:
----------
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
      CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ11 2010-Feb-11, SIN-BKK 2010-Feb-11, 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-11, SIN-BKK 2010-Feb-11, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-11, SIN-BKK 2010-Feb-11, Y, 2, 0, 0, 0, 0, 300, 0,
****************************************
****************************************
Subclasses:
```

```
-----------
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
      GrpBks (pdg), StfBkgs, WLBkgs, 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, 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, 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,
SQ11 2010-Feb-18, SIN-BKK, 2010-Feb-18, 08:20:00, 2010-Feb-18, 11:00:00, 07:40:
      00, 0, -05:00:00, 6300, 0,
****************************************
****************************************
LegCabins:
----------
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
      CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ11 2010-Feb-18, SIN-BKK 2010-Feb-18, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300
      , 9, 0, 0, 0, 0, 0,
****************************************
****************************************
Buckets:
--------
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
****************************************
****************************************
SegmentCabins:
--------------
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-18, SIN-BKK 2010-Feb-18, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-18, SIN-BKK 2010-Feb-18, Y, 2, 0, 0, 0, 0, 300, 0,
****************************************
****************************************
Subclasses:
-----------
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
      GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQ11 2010-Feb-18, SIN-BKK 2010-Feb-18, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
SQ11 2010-Feb-18, SIN-BKK 2010-Feb-18, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
****************************************
****************************************
```

```
FlightDate: SQ11, 2010-Feb-19
****************************************
****************************************
Leg-Dates:
----------
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
      Elapsed, Distance, Capacity,
SQ11 2010-Feb-19, SIN-BKK, 2010-Feb-19, 08:20:00, 2010-Feb-19, 11:00:00, 07:40:
      00, 0, -05:00:00, 6300, 0,
****************************************
****************************************
LegCabins:
----------
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
      CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ11 2010-Feb-19, SIN-BKK 2010-Feb-19, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300
      , 9, 0, 0, 0, 0, 0,
****************************************
****************************************
Buckets:
--------
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
****************************************
****************************************
SegmentCabins:
--------------
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-19, SIN-BKK 2010-Feb-19, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-19, SIN-BKK 2010-Feb-19, Y, 2, 0, 0, 0, 0, 300, 0,
****************************************
****************************************
Subclasses:
-----------
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
      GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQ11 2010-Feb-19, SIN-BKK 2010-Feb-19, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
SQ11 2010-Feb-19, SIN-BKK 2010-Feb-19, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
****************************************
****************************************
FlightDate: SQ11, 2010-Feb-20
****************************************
****************************************
Leg-Dates:
----------
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
      Elapsed, Distance, Capacity,
SQ11 2010-Feb-20, SIN-BKK, 2010-Feb-20, 08:20:00, 2010-Feb-20, 11:00:00, 07:40:
      00, 0, -05:00:00, 6300, 0,
****************************************
****************************************
LegCabins:
----------
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
      CommSpace, AvPool, Avl, NAV, 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, 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, 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, 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, 300
      , 9, 0, 0, 0, 0, 0,
****************************************
****************************************
Buckets:
--------
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
****************************************
****************************************
SegmentCabins:
--------------
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-26, SIN-BKK 2010-Feb-26, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-26, SIN-BKK 2010-Feb-26, Y, 2, 0, 0, 0, 0, 300, 0,
****************************************
****************************************
Subclasses:
-----------
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
      GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQ11 2010-Feb-26, SIN-BKK 2010-Feb-26, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
SQ11 2010-Feb-26, SIN-BKK 2010-Feb-26, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
****************************************
****************************************
FlightDate: SQ11, 2010-Feb-27
****************************************
****************************************
Leg-Dates:
----------
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
      Elapsed, Distance, Capacity,
SQ11 2010-Feb-27, SIN-BKK, 2010-Feb-27, 08:20:00, 2010-Feb-27, 11:00:00, 07:40:
      00, 0, -05:00:00, 6300, 0,
****************************************
****************************************
LegCabins:
----------
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
      CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ11 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,
SQ11 2010-Feb-27, SIN-BKK 2010-Feb-27, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-27, SIN-BKK 2010-Feb-27, Y, 2, 0, 0, 0, 0, 300, 0,
****************************************
****************************************
Subclasses:
-----------
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
      GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQ11 2010-Feb-27, SIN-BKK 2010-Feb-27, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
SQ11 2010-Feb-27, SIN-BKK 2010-Feb-27, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
****************************************
****************************************
FlightDate: SQ11, 2010-Feb-28
```

```
****************************************
****************************************
Leg-Dates:
----------
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
      Elapsed, Distance, Capacity,
SQ11 2010-Feb-28, SIN-BKK, 2010-Feb-28, 08:20:00, 2010-Feb-28, 11:00:00, 07:40:
      00, 0, -05:00:00, 6300, 0,
****************************************
****************************************
LegCabins:
----------
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
      CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ11 2010-Feb-28, SIN-BKK 2010-Feb-28, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300
      , 9, 0, 0, 0, 0, 0,
****************************************
****************************************
Buckets:
--------
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
****************************************
****************************************
SegmentCabins:
--------------
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-28, SIN-BKK 2010-Feb-28, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-28, SIN-BKK 2010-Feb-28, Y, 2, 0, 0, 0, 0, 300, 0,
****************************************
****************************************
Subclasses:
-----------
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
      GrpBks (pdg), StfBkgs, WLBkgs, 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, 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, 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, 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, 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, 0, 0, 200
      , 9, 0, 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, 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, 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, 200, 0,
SQ12 2010-Jan-23, SIN-HND 2010-Jan-23, Y, 2, 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, 200, 0,
SQ12 2010-Jan-24, SIN-HND 2010-Jan-24, Y, 2, 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,
SQ12 2010-Jan-25, SIN-HND 2010-Jan-25, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200
      , 9, 0, 0, 0, 0, 0,
****************************************
****************************************
Buckets:
--------
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
****************************************
****************************************
SegmentCabins:
--------------
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Jan-25, SIN-HND 2010-Jan-25, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Jan-25, SIN-HND 2010-Jan-25, Y, 2, 0, 0, 0, 0, 200, 0,
****************************************
****************************************
Subclasses:
-----------
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
      GrpBks (pdg), StfBkgs, WLBkgs, ETB, ClassAvl, RevAvl, SegAvl,
SQ12 2010-Jan-25, SIN-HND 2010-Jan-25, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
SQ12 2010-Jan-25, SIN-HND 2010-Jan-25, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
****************************************
****************************************
FlightDate: SQ12, 2010-Jan-26
****************************************
****************************************
Leg-Dates:
----------
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
      Elapsed, Distance, Capacity,
SQ12 2010-Jan-26, SIN-HND, 2010-Jan-26, 09:20:00, 2010-Jan-26, 12:00:00, 07:40:
      00, 0, -05:00:00, 6300, 0,
****************************************
****************************************
LegCabins:
----------
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
      CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ12 2010-Jan-26, SIN-HND 2010-Jan-26, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200
      , 9, 0, 0, 0, 0, 0,
****************************************
****************************************
Buckets:
--------
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
****************************************
****************************************
SegmentCabins:
--------------
```

```
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Jan-26, SIN-HND 2010-Jan-26, Y, 1, 0, 0, 0, 200, 0,
SQ12 2010-Jan-26, SIN-HND 2010-Jan-26, Y, 2, 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-26, SIN-HND 2010-Jan-26, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
SQ12 2010-Jan-26, SIN-HND 2010-Jan-26, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0,
      0, 0, 0, 0, 0,
****************************************
****************************************
FlightDate: SQ12, 2010-Jan-27
****************************************
****************************************
Leg-Dates:
----------
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset,
      Elapsed, Distance, Capacity,
SQ12 2010-Jan-27, SIN-HND, 2010-Jan-27, 09:20:00, 2010-Jan-27, 12:00:00, 07:40:
      00, 0, -05:00:00, 6300, 0,
****************************************
****************************************
LegCabins:
----------
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group,
      CommSpace, AvPool, Avl, NAV, GAV, ACP, ETB, BidPrice,
SQ12 2010-Jan-27, SIN-HND 2010-Jan-27, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200
      , 9, 0, 0, 0, 0, 0,
****************************************
****************************************
Buckets:
--------
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
****************************************
****************************************
SegmentCabins:
--------------
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Jan-27, SIN-HND 2010-Jan-27, Y, 1, 0, 0, 0, 200, 0,
SQ12 2010-Jan-27, SIN-HND 2010-Jan-27, Y, 2, 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, 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, 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, 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, 0, 200
      , 9, 0, 0, 0, 0, 0,
****************************************
****************************************
```

```
Buckets:
--------
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
****************************************
****************************************
SegmentCabins:
--------------
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-06, SIN-HND 2010-Feb-06, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-06, SIN-HND 2010-Feb-06, Y, 2, 0, 0, 0, 0, 200, 0,
****************************************
****************************************
Subclasses:
-----------
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
      GrpBks (pdg), StfBkgs, WLBkgs, 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, 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, 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, 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, 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, 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, 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, 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, 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, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-21, SIN-HND 2010-Feb-21, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-21, SIN-HND 2010-Feb-21, Y, 2, 0, 0, 0, 0, 200, 0,
****************************************
****************************************
Subclasses:
-----------
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs,
      GrpBks (pdg), StfBkgs, WLBkgs, 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, 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, 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, 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, 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, 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, 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, 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

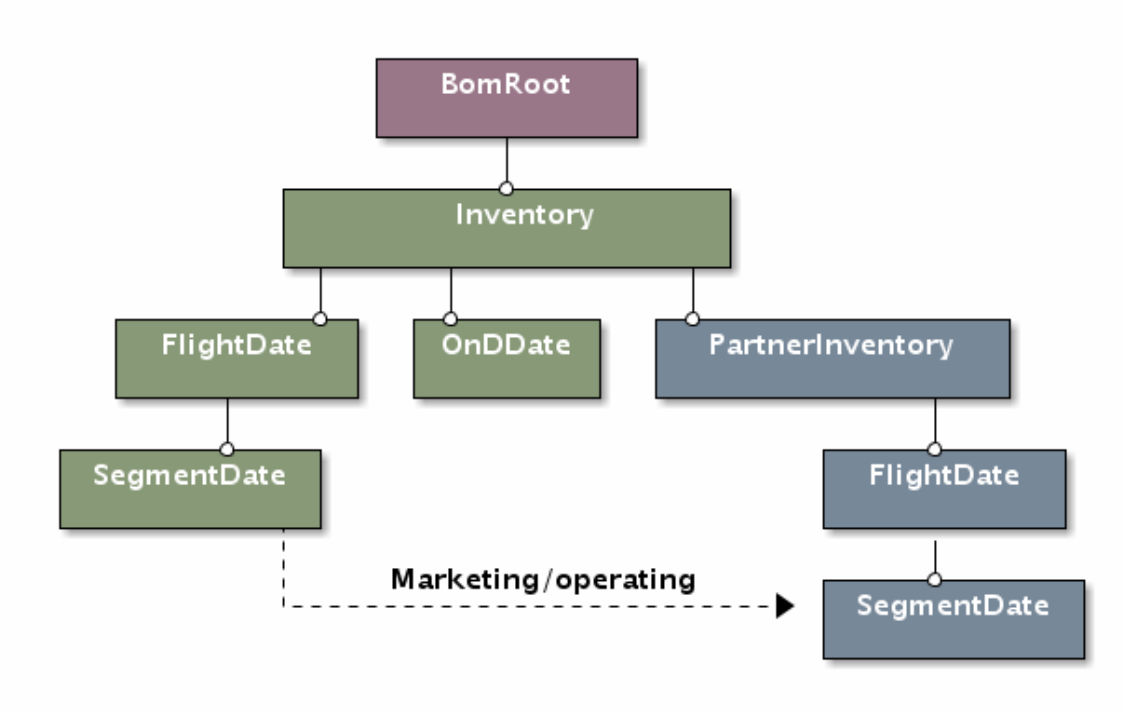


Figure 2: SEvMgr BOM tree

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

12.6.1 Airline Network BOM Tree

- `SEVMGR::ReachableUniverse`
- `SEVMGR::OriginDestinationSet`
- `SEVMGR::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 SEvMgr aims at calculating a list of `travel solutions` for every incoming `booking request`.

13 Supported Systems

13.1 Table of Contents

- Introduction
- .1 SEvMgr 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 SEvMgr 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 SEvMgr External
    - Microsoft Windows XP with MS Visual C++ and Intel MKL
  - Unix Systems
    - SunOS 5.9 with SEvMgr External
- SEvMgr 3.9.1
- SEvMgr 3.9.0
- SEvMgr 3.8.1

13.2 Introduction

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

14 SEvMgr Supported Systems (Previous Releases)

14.1 SEvMgr 3.9.1

14.2 SEvMgr 3.9.0

14.3 SEvMgr 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 instanciate the AIRSCHED_Service object:

Then, we construct a default sample list of travel solutions and a default booking request (as mentionned in ug_procedure_bookingrequest and ug_procedure_-travelsolution parts):

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::CmdBom-Manager.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 builde 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 Bristish 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 Use Sevmgr elements

```
 */
// ////////////////////////////////////////////////////////////////////
// Import section
// ////////////////////////////////////////////////////////////////////
// STL
#include <sstream>
#include <fstream>
#include <map>
#include <cmath>
// Boost Unit Test Framework (UTF)
#define BOOST_TEST_DYN_LINK
#define BOOST_TEST_MAIN
#define BOOST_TEST_MODULE EventQueueManagementTest
#include <boost/test/unit_test.hpp>
// StdAir
#include <stdair/stdair_basic_types.hpp>
#include <stdair/basic/BasLogParams.hpp>
#include <stdair/basic/BasDBParams.hpp>
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/basic/ProgressStatusSet.hpp>
#include <stdair/bom/EventStruct.hpp>
#include <stdair/bom/EventQueue.hpp>
#include <stdair/bom/BookingRequestStruct.hpp>
#include <stdair/service/Logger.hpp>
// SEvMgr
#include <sevmgr/SEVMGR_Service.hpp>
#include <sevmgr/config/sevmgr-paths.hpp>

namespace boost_utf = boost::unit_test;
```

```
// (Boost) Unit Test XML Report
std::ofstream utfReportStream ("EventQueueManagementTestSuite_utfresults.xml");

struct UnitTestConfig {
  UnitTestConfig() {
    boost_utf::unit_test_log.set_stream (utfReportStream);
    boost_utf::unit_test_log.set_format (boost_utf::XML);
    boost_utf::unit_test_log.set_threshold_level (boost_utf::log_test_units);
    //boost_utf::unit_test_log.set_threshold_level
       (boost_utf::log_successful_tests);
  }

  ~UnitTestConfig() {
  }
};

// Specific type definitions
typedef std::pair<stdair::Count_T, stdair::Count_T> NbOfEventsPair_T;
typedef std::map<const stdair::DemandStreamKeyStr_T,
                 NbOfEventsPair_T> NbOfEventsByDemandStreamMap_T;


// ///////////////// 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 (sevmgr_simple_simulation_test) {

  // Input file name
  const stdair::Filename_T lInputFilename (STDAIR_SAMPLE_DIR "/demand01.csv");

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

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

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

  // Initialise the Sevmgr service object
  const stdair::BasLogParams lLogParams (stdair::LOG::DEBUG, logOutputFile);
  SEVMGR::SEVMGR_Service sevmgrService (lLogParams);

  sevmgrService.buildSampleBom();

  sevmgrService.reset();

  // DEBUG
  STDAIR_LOG_DEBUG ("End of the simulation");

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

// End the test suite
BOOST_AUTO_TEST_SUITE_END()

/*!
```

17 Directory Hierarchy

17.1 Directories

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

sevmgr 112
basic 110
batches 111
command 111
config 111
factory 111
python 112
service 112
ui 113
cmdline 111
test 112
sevmgr 112

18 Namespace Index

18.1 Namespace List

Here is a list of all namespaces with brief descriptions:

SEVMGR 113
stdair 116
swift
The wrapper namespace 116

19 Class Index

19.1 Class Hierarchy

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

std::allocator 116

std::auto_ptr 116

std::basic_fstream< char > 118

std::fstream 133

std::basic_fstream< wchar_t > 118

std::wfstream 164

std::basic_ifstream< char > 118

std::ifstream 133

std::basic_ifstream< wchar_t > 118

std::wifstream 164

std::basic_ios< char > 119

std::ios 134

std::basic_ios< Char > 119

std::basic_istream 120

std::basic_ostream 121

std::basic_ios< wchar_t > 119

std::wios 165

std::basic_iostream< Char > 119

std::basic_fstream 118

std::basic_stringstream 122

std::basic_istream< Char > 120

std::basic_ifstream 118

std::basic_iostream 119

std::basic_istringstream 120

std::basic_istream< char > 120

std::istream 135

std::basic_istream< wchar_t > 120

std::wistream 165

std::basic_istringstream< char > 120

std::istringstream 135

std::basic_istringstream< wchar_t > 120

std::wistringstream 165

std::basic_ofstream< char > 120

std::ofstream 141

std::basic_ofstream< wchar_t > 120

std::wofstream 166

std::basic_ostream< char > 121

std::ostream 141

std::basic_ostream< Char > 121

std::basic_iostream 119

std::basic_ofstream 120

std::basic_ostringstream 121

std::basic_ostream< wchar_t > 121

std::wostream 166

std::basic_ostringstream< char > 121

std::ostringstream 142

std::basic_ostringstream< wchar_t > 121

std::wostringstream 167

std::basic_string 122

std::basic_string< char > 122

std::string 162

std::basic_string< wchar_t > 122

std::wstring 167

std::basic_stringstream< char > 122

std::stringstream 162

std::basic_stringstream< wchar_t > 122

std::wstringstream 168

std::bitset 122

CmdAbstract 123

SEVMGR::EventQueueManager 129

COMMAND 123

std::complex 124

std::list::const_iterator 124

std::map::const_iterator 124

std::multimap::const_iterator 125

std::set::const_iterator 125

std::multiset::const_iterator 125

std::basic_string::const_iterator 125

std::vector::const_iterator 125

std::string::const_iterator 126

std::wstring::const_iterator 126

std::deque::const_iterator 126

std::map::const_reverse_iterator 126

std::list::const_reverse_iterator 127

std::multimap::const_reverse_iterator 127

std::set::const_reverse_iterator 127

std::multiset::const_reverse_iterator 127

std::basic_string::const_reverse_iterator 127

std::vector::const_reverse_iterator 128

std::string::const_reverse_iterator 128

std::wstring::const_reverse_iterator 128

std::deque::const_reverse_iterator 128

std::deque 129

std::exception 130

- std::bad_alloc 116
- std::bad_cast 117
- std::bad_exception 117
- std::bad_typeid 118
- std::ios_base::failure 133
- std::logic_error 139
  - std::domain_error 129
  - std::invalid_argument 134
  - std::length_error 138
  - std::out_of_range 142
- std::runtime_error 148
  - std::overflow_error 143
  - std::range_error 145
  - std::underflow_error 163

FacServiceAbstract 131

- SEVMGR::FacSEVMGRServiceContext 131

std::ios_base 135

- std::basic_ios 119

std::list::iterator 136

std::deque::iterator 136

std::multimap::iterator 136

std::set::iterator 137

std::multiset::iterator 137

std::basic_string::iterator 137

std::vector::iterator 137

std::string::iterator 137

std::map::iterator 138

std::wstring::iterator 138

std::list 139

std::map 139

std::multimap 140

std::multiset 140

std::priority_queue 143

SEVMGR::PYEventQueueManager 143

std::queue 144

std::vector::reverse_iterator 145

std::list::reverse_iterator 145

std::map::reverse_iterator 146

std::deque::reverse_iterator 146

std::wstring::reverse_iterator 146

std::basic_string::reverse_iterator 146

std::multimap::reverse_iterator 146

std::set::reverse_iterator 147

std::multiset::reverse_iterator 147

std::string::reverse_iterator 147

RootException 147

SEVMGR::SEvMgrException 153

ServiceAbstract 148

SEVMGR::SEVMGR_ServiceContext 152

std::set 148

SEVMGR::SEVMGR_Service 149

swift::SKeymap 154

swift::SReadline 156

std::stack 161

std::valarray 163

std::vector 163

20 Class Index

20.1 Class List

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

std::allocator
STL class 116

std::auto_ptr
STL class 116

std::bad_alloc
STL class 116

std::bad_cast
STL class 117

std::bad_exception
STL class 117

std::bad_typeid
STL class 118

std::basic_fstream
STL class 118

std::basic_ifstream
STL class 118

std::basic_ios
STL class 119

std::basic_iostream
STL class 119

std::basic_istream
STL class 120

std::basic_istringstream
STL class 120

std::basic_ofstream
STL class 120

std::basic_ostream
STL class 121

std::basic_ostringstream
STL class 121

std::basic_string
STL class 122

std::basic_stringstream
STL class 122

std::bitset
STL class 122

CmdAbstract 123

COMMAND 123

std::complex
STL class 124

std::list::const_iterator
STL iterator class 124

std::map::const_iterator
STL iterator class 124

std::multimap::const_iterator
STL iterator class 125

std::set::const_iterator
STL iterator class 125

std::multiset::const_iterator
STL iterator class 125

std::basic_string::const_iterator
STL iterator class 125

std::vector::const_iterator
STL iterator class 125

std::string::const_iterator
STL iterator class 126

std::wstring::const_iterator
STL iterator class 126

std::deque::const_iterator
STL iterator class 126

std::map::const_reverse_iterator
STL iterator class 126

std::list::const_reverse_iterator
STL iterator class 127

std::multimap::const_reverse_iterator
STL iterator class 127

std::set::const_reverse_iterator
STL iterator class 127

std::multiset::const_reverse_iterator
STL iterator class 127

std::basic_string::const_reverse_iterator
STL iterator class 127

std::vector::const_reverse_iterator
STL iterator class 128

std::string::const_reverse_iterator
STL iterator class 128

std::wstring::const_reverse_iterator
STL iterator class 128

std::deque::const_reverse_iterator
STL iterator class 128

std::deque
STL class 129

std::domain_error
STL class 129

SEVMGR::EventQueueManager
Utility class for Demand and DemandStream objects 129

std::exception
STL class 130

FacServiceAbstract 131

SEVMGR::FacSEVMGRServiceContext 131

std::ios_base::failure
STL class 133

std::fstream
STL class 133

std::ifstream
STL class 133

std::invalid_argument
STL class 134

std::ios
STL class 134

std::ios_base
STL class 135

std::istream
STL class 135

std::istringstream
STL class 135

std::list::iterator
STL iterator class 136

std::deque::iterator
STL iterator class 136

std::multimap::iterator
STL iterator class 136

std::set::iterator
STL iterator class 137

std::multiset::iterator
STL iterator class 137

std::basic_string::iterator
STL iterator class 137

std::vector::iterator
STL iterator class 137

std::string::iterator
STL iterator class 137

std::map::iterator
STL iterator class 138

std::wstring::iterator
STL iterator class 138

std::length_error
STL class 138

std::list
STL class 139

std::logic_error
STL class 139

std::map
STL class 139

std::multimap
STL class 140

std::multiset
STL class 140

std::ofstream
STL class 141

std::ostream
STL class 141

std::ostringstream
STL class 142

std::out_of_range
STL class 142

std::overflow_error
STL class 143

std::priority_queue
STL class 143

SEVMGR::PYEventQueueManager 143

std::queue
STL class 144

std::range_error
STL class 145

std::vector::reverse_iterator
STL iterator class 145

std::list::reverse_iterator
STL iterator class 145

std::map::reverse_iterator
STL iterator class 146

std::deque::reverse_iterator
STL iterator class 146

std::wstring::reverse_iterator
STL iterator class 146

std::basic_string::reverse_iterator
STL iterator class 146

std::multimap::reverse_iterator
STL iterator class 146

std::set::reverse_iterator
STL iterator class 147

std::multiset::reverse_iterator
STL iterator class 147

std::string::reverse_iterator
STL iterator class 147

RootException 147

std::runtime_error
STL class 148

ServiceAbstract 148

std::set
STL class 148

SEVMGR::SEVMGR_Service
Class holding the services related to Travel Demand Generation 149

SEVMGR::SEVMGR_ServiceContext
Class holding the context of the Sevmgr services 152

SEVMGR::SEvMgrException 153

swift::SKeymap
The readline keymap wrapper 154

swift::SReadline
The readline library wrapper 156

std::stack
STL class 161

std::string
STL class 162

std::stringstream
STL class 162

std::underflow_error
STL class 163

std::valarray
STL class 163

std::vector
STL class 163

std::wfstream
STL class 164

std::wifstream
STL class 164

std::wios
STL class 165

std::wistream
STL class 165

std::wistringstream
STL class 165

std::wofstream
STL class 166

std::wostream
STL class 166

std::wostringstream
STL class 167

std::wstring
STL class 167

std::wstringstream
STL class 168

21 File Index

21.1 File List

Here is a list of all files with brief descriptions:

sevmgr/SEVMGR_Exceptions.hpp 194

sevmgr/SEVMGR_Service.hpp 195

sevmgr/SEVMGR_Types.hpp 196

sevmgr/basic/BasConst.cpp 169

sevmgr/basic/BasConst_EventQueueManager.hpp 169

sevmgr/basic/BasConst_SEVMGR_Service.hpp 170

sevmgr/basic/BasParserTypes.hpp 171

sevmgr/batches/sevmgr_demo.cpp 174

sevmgr/command/EventQueueManager.cpp 176

sevmgr/command/EventQueueManager.hpp 177

sevmgr/config/sevmgr-paths.hpp 179

sevmgr/config/sevmgr-paths.hpp.in 182

sevmgr/factory/FacSEVMGRServiceContext.cpp 182

sevmgr/factory/FacSEVMGRServiceContext.hpp 183

sevmgr/python/pysevmgr.cpp 185

sevmgr/service/SEVMGR_Service.cpp 187

sevmgr/service/SEVMGR_ServiceContext.cpp 191

sevmgr/service/SEVMGR_ServiceContext.hpp 192

sevmgr/ui/cmdline/readline_autocomp.hpp 201

sevmgr/ui/cmdline/sevmgr.cpp 205

sevmgr/ui/cmdline/SReadline.hpp
C++ wrapper around libreadline 217

test/sevmgr/EventQueueManagementTestSuite.cpp 223

22 Directory Documentation

22.1 sevmgr/basic/ Directory Reference

Files

- file BasConst.cpp
- file BasConst_EventQueueManager.hpp
- file BasConst_SEVMGR_Service.hpp
- file BasParserTypes.hpp

22.2 sevmgr/batches/ Directory Reference

Files

- file sevmgr_demo.cpp

22.3 sevmgr/ui/cmdline/ Directory Reference

Files

- file readline_autocomp.hpp
- file sevmgr.cpp
- file SReadline.hpp

  C++ wrapper around libreadline.

22.4 sevmgr/command/ Directory Reference

Files

- file EventQueueManager.cpp
- file EventQueueManager.hpp

22.5 sevmgr/config/ Directory Reference

Files

- file sevmgr-paths.hpp
- file sevmgr-paths.hpp.in

22.6 sevmgr/factory/ Directory Reference

Files

- file FacSEVMGRServiceContext.cpp
- file FacSEVMGRServiceContext.hpp

22.7 sevmgr/python/ Directory Reference

Files

- file pysevmgr.cpp

22.8 sevmgr/service/ Directory Reference

Files

- file SEVMGR_Service.cpp
- file SEVMGR_ServiceContext.cpp
- file SEVMGR_ServiceContext.hpp

22.9 test/sevmgr/ Directory Reference

Files

- file EventQueueManagementTestSuite.cpp

22.10 sevmgr/ Directory Reference

Directories

- directory basic
- directory batches
- directory command
- directory config
- directory factory
- directory python
- directory service
- directory ui

Files

- file SEVMGR_Exceptions.hpp
- file SEVMGR_Service.hpp
- file SEVMGR_Types.hpp

22.11 test/ Directory Reference

Directories

- directory sevmgr

22.12 sevmgr/ui/ Directory Reference

Directories

- directory cmdline

23 Namespace Documentation

23.1 SEVMGR Namespace Reference

Classes

- class EventQueueManager

  Utility class for Demand and DemandStream objects.
- class FacSEVMGRServiceContext
- struct PYEventQueueManager
- class SEVMGR_ServiceContext

  Class holding the context of the Sevmgr services.
- class SEvMgrException
- class SEVMGR_Service

  class holding the services related to Travel Demand Generation.

Typedefs

- typedef char char_t
- typedef boost::spirit::classic::file_iterator < char_t > iterator_t
- typedef boost::spirit::classic::scanner < iterator_t > scanner_t
- typedef boost::spirit::classic::rule < scanner_t > rule_t
- typedef boost::spirit::classic::int_parser < unsigned int, 10, 1, 1 > int1_p_t
- typedef boost::spirit::classic::uint_parser < unsigned int, 10, 2, 2 > uint2_p_t
- typedef boost::spirit::classic::uint_parser < unsigned int, 10, 1, 2 > uint1_2_p_t
- typedef boost::spirit::classic::uint_parser < unsigned int, 10, 1, 3 > uint1_3_p_t
- typedef boost::spirit::classic::uint_parser < unsigned int, 10, 4, 4 > uint4_p_t
- typedef boost::spirit::classic::uint_parser < unsigned int, 10, 1, 4 > uint1_4_p_t
- typedef boost::spirit::classic::chset < char_t > chset_t
- typedef boost::spirit::classic::impl::loop_traits < chset_t, unsigned int, unsigned int >::type repeat_p_t
- typedef boost::spirit::classic::bounded < uint2_p_t, unsigned int > bounded2_p_t
- typedef boost::spirit::classic::bounded < uint1_2_p_t, unsigned int > bounded1_2_p_t
- typedef boost::spirit::classic::bounded < uint1_3_p_t, unsigned int > bounded1_3_p_t
- typedef boost::spirit::classic::bounded < uint4_p_t, unsigned int > bounded4_p_t

- typedef boost::spirit::classic::bounded < uint1_4_p_t, unsigned int > bounded1-_4_p_t
- typedef boost::shared_ptr < SEVMGR_Service > SEVMGR_ServicePtr_T

23.1.1 Typedef Documentation

23.1.1.1 typedef char SEVMGR::char_t

Definition at line 31 of file BasParserTypes.hpp.

23.1.1.2 typedef boost::spirit::classic::file_iterator<char_t> SEVMGR::iterator_t

Definition at line 35 of file BasParserTypes.hpp.

23.1.1.3 typedef boost::spirit::classic::scanner<iterator_t> SEVMGR::scanner_t

Definition at line 36 of file BasParserTypes.hpp.

23.1.1.4 typedef boost::spirit::classic::rule<scanner_t> SEVMGR::rule_t

Definition at line 37 of file BasParserTypes.hpp.

23.1.1.5 typedef boost::spirit::classic::int_parser<unsigned int, 10, 1, 1> SEVMGR::int1_p_t

1-digit-integer parser

Definition at line 45 of file BasParserTypes.hpp.

23.1.1.6 typedef boost::spirit::classic::uint_parser<unsigned int, 10, 2, 2> SEVMGR::uint2_p_t

2-digit-integer parser

Definition at line 48 of file BasParserTypes.hpp.

23.1.1.7 typedef boost::spirit::classic::uint_parser<unsigned int, 10, 1, 2> SEVMGR::uint1_2_p_t

Up-to-2-digit-integer parser

Definition at line 51 of file BasParserTypes.hpp.

23.1.1.8 typedef boost::spirit::classic::uint_parser<unsigned int, 10, 1, 3> SEVMGR::uint1_3_p_t

Up-to-3-digit-integer parser

Definition at line 54 of file BasParserTypes.hpp.

23.1.1.9 typedef boost::spirit::classic::uint_parser<unsigned int, 10, 4, 4> SEVMGR::uint4_p_t

4-digit-integer parser

Definition at line 57 of file BasParserTypes.hpp.

23.1.1.10 typedef boost::spirit::classic::uint_parser<unsigned int, 10, 1, 4> SEVMGR::uint1_4_p_t

Up-to-4-digit-integer parser

Definition at line 60 of file BasParserTypes.hpp.

23.1.1.11 typedef boost::spirit::classic::chset<char_t> SEVMGR::chset_t

character set

Definition at line 63 of file BasParserTypes.hpp.

23.1.1.12 typedef boost::spirit::classic::impl::loop_traits<chset_t, unsigned int, unsigned int>::type SEVMGR::repeat_p_t

(Repeating) sequence of a given number of characters: repeat_p(min, max)

Definition at line 69 of file BasParserTypes.hpp.

23.1.1.13 typedef boost::spirit::classic::bounded<uint2_p_t, unsigned int> SEVMGR::bounded2_p_t

Bounded-number-of-integers parser

Definition at line 72 of file BasParserTypes.hpp.

23.1.1.14 typedef boost::spirit::classic::bounded<uint1_2_p_t, unsigned int> SEVMGR::bounded1_2_p_t

Definition at line 73 of file BasParserTypes.hpp.

23.1.1.15 typedef boost::spirit::classic::bounded<uint1_3_p_t, unsigned int> SEVMGR::bounded1_3_p_t

Definition at line 74 of file BasParserTypes.hpp.

23.1.1.16 typedef boost::spirit::classic::bounded<uint4_p_t, unsigned int> SEVMGR::bounded4_p_t

Definition at line 75 of file BasParserTypes.hpp.

23.1.1.17 typedef boost::spirit::classic::bounded<uint1_4_p_t, unsigned int> SEVMGR::bounded1_4_p_t

Definition at line 76 of file BasParserTypes.hpp.

23.1.1.18 typedef boost::shared_ptr<SEVMGR_Service> SEVMGR::SEVMGR_ServicePtr_T

(Smart) Pointer on the SEvMgr service handler.

Definition at line 15 of file SEVMGR_Types.hpp.

23.2 stdair Namespace Reference

23.3 swift Namespace Reference

The wrapper namespace.

Classes

- class SKeymap

  The readline keymap wrapper.
- class SReadline

  The readline library wrapper.

23.3.1 Detailed Description

The wrapper namespace. The namespace is also used for other library elements.

24 Class Documentation

24.1 std::allocator Class Reference

STL class.

24.1.1 Detailed Description

STL class.

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

24.2 std::auto_ptr Class Reference

STL class.

24.2.1 Detailed Description

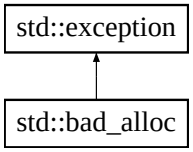
STL class.

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

24.3 std::bad_alloc Class Reference

STL class.

Inheritance diagram for std::bad_alloc:



24.3.1 Detailed Description

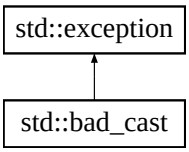
STL class.

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

24.4 std::bad_cast Class Reference

STL class.

Inheritance diagram for std::bad_cast:



24.4.1 Detailed Description

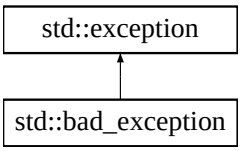
STL class.

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

24.5 std::bad_exception Class Reference

STL class.

Inheritance diagram for std::bad_exception:



24.5.1 Detailed Description

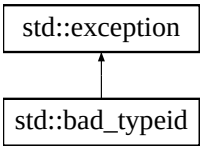
STL class.

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

24.6 std::bad_typeid Class Reference

STL class.

Inheritance diagram for std::bad_typeid:



24.6.1 Detailed Description

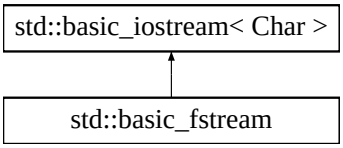
STL class.

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

24.7 std::basic_fstream Class Reference

STL class.

Inheritance diagram for std::basic_fstream:



24.7.1 Detailed Description

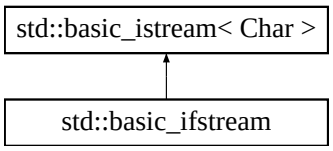
STL class.

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

24.8 std::basic_ifstream Class Reference

STL class.

Inheritance diagram for std::basic_ifstream:



24.8.1 Detailed Description

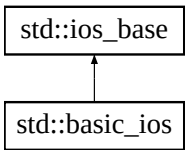
STL class.

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

24.9 std::basic_ios Class Reference

STL class.

Inheritance diagram for std::basic_ios:



24.9.1 Detailed Description

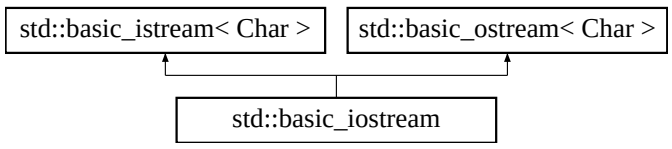
STL class.

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

24.10 std::basic_iostream Class Reference

STL class.

Inheritance diagram for std::basic_iostream:



24.10.1 Detailed Description

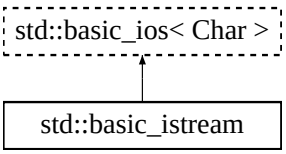
STL class.

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

24.11 std::basic_istream Class Reference

STL class.

Inheritance diagram for std::basic_istream:



24.11.1 Detailed Description

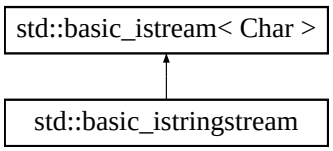
STL class.

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

24.12 std::basic_istringstream Class Reference

STL class.

Inheritance diagram for std::basic_istringstream:



24.12.1 Detailed Description

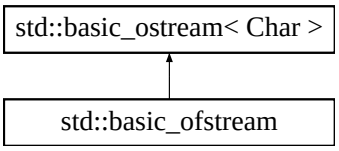
STL class.

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

24.13 std::basic_ofstream Class Reference

STL class.

Inheritance diagram for std::basic_ofstream:



24.13.1 Detailed Description

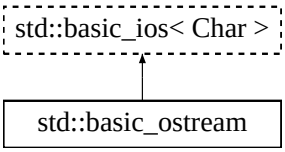
STL class.

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

24.14 std::basic_ostream Class Reference

STL class.

Inheritance diagram for std::basic_ostream:



24.14.1 Detailed Description

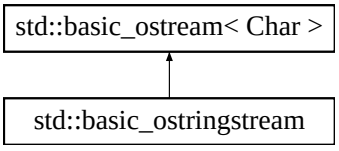
STL class.

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

24.15 std::basic_ostringstream Class Reference

STL class.

Inheritance diagram for std::basic_ostringstream:



24.15.1 Detailed Description

STL class.

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

24.16 std::basic_string Class Reference

STL class.

Classes

- class const_iterator

  STL iterator class.
- class const_reverse_iterator

  STL iterator class.
- class iterator

  STL iterator class.
- class reverse_iterator

  STL iterator class.

24.16.1 Detailed Description

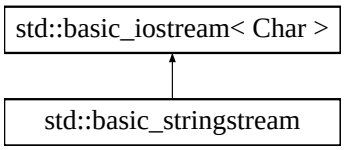
STL class.

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

24.17 std::basic_stringstream Class Reference

STL class.

Inheritance diagram for std::basic_stringstream:



24.17.1 Detailed Description

STL class.

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

24.18 std::bitset Class Reference

STL class.

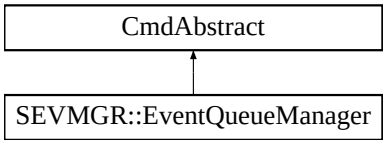
24.18.1 Detailed Description

STL class.

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

24.19 CmdAbstract Class Reference

Inheritance diagram for CmdAbstract:



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

- sevmgr/command/EventQueueManager.hpp

24.20 COMMAND Struct Reference

`#include <sevmgr/ui/cmdline/readline_autocomp.hpp>`

Public Attributes

- char const * name
- pt2Func * func
- char * doc

24.20.1 Detailed Description

A structure which contains information on the commands this program can understand.

24.20.2 Member Data Documentation

24.20.2.1 char const* COMMAND::name

User printable name of the function.

Definition at line 45 of file readline_autocomp.hpp.

Referenced by find_command(), and com_help().

24.20.2.2 pt2Func* COMMAND::func

Function to call to do the job.

Definition at line 50 of file readline_autocomp.hpp.

Referenced by execute_line().

24.20.2.3 char* COMMAND::doc

Documentation for this function.

Definition at line 55 of file readline_autocomp.hpp.

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

- sevmgr/ui/cmdline/readline_autocomp.hpp

24.21 std::complex Class Reference

STL class.

24.21.1 Detailed Description

STL class.

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

24.22 std::list::const_iterator Class Reference

STL iterator class.

24.22.1 Detailed Description

STL iterator class.

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

24.23 std::map::const_iterator Class Reference

STL iterator class.

24.23.1 Detailed Description

STL iterator class.

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

24.24 std::multimap::const_iterator Class Reference

STL iterator class.

24.24.1 Detailed Description

STL iterator class.

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

24.25 std::set::const_iterator Class Reference

STL iterator class.

24.25.1 Detailed Description

STL iterator class.

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

24.26 std::multiset::const_iterator Class Reference

STL iterator class.

24.26.1 Detailed Description

STL iterator class.

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

24.27 std::basic_string::const_iterator Class Reference

STL iterator class.

24.27.1 Detailed Description

STL iterator class.

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

24.28 std::vector::const_iterator Class Reference

STL iterator class.

24.28.1 Detailed Description

STL iterator class.

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

24.29 std::string::const_iterator Class Reference

STL iterator class.

24.29.1 Detailed Description

STL iterator class.

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

24.30 std::wstring::const_iterator Class Reference

STL iterator class.

24.30.1 Detailed Description

STL iterator class.

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

24.31 std::deque::const_iterator Class Reference

STL iterator class.

24.31.1 Detailed Description

STL iterator class.

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

24.32 std::map::const_reverse_iterator Class Reference

STL iterator class.

24.32.1 Detailed Description

STL iterator class.

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

24.33 std::list::const_reverse_iterator Class Reference

STL iterator class.

24.33.1 Detailed Description

STL iterator class.

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

24.34 std::multimap::const_reverse_iterator Class Reference

STL iterator class.

24.34.1 Detailed Description

STL iterator class.

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

24.35 std::set::const_reverse_iterator Class Reference

STL iterator class.

24.35.1 Detailed Description

STL iterator class.

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

24.36 std::multiset::const_reverse_iterator Class Reference

STL iterator class.

24.36.1 Detailed Description

STL iterator class.

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

24.37 std::basic_string::const_reverse_iterator Class Reference

STL iterator class.

24.37.1 Detailed Description

STL iterator class.

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

24.38 std::vector::const_reverse_iterator Class Reference

STL iterator class.

24.38.1 Detailed Description

STL iterator class.

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

24.39 std::string::const_reverse_iterator Class Reference

STL iterator class.

24.39.1 Detailed Description

STL iterator class.

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

24.40 std::wstring::const_reverse_iterator Class Reference

STL iterator class.

24.40.1 Detailed Description

STL iterator class.

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

24.41 std::deque::const_reverse_iterator Class Reference

STL iterator class.

24.41.1 Detailed Description

STL iterator class.

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

24.42 std::deque Class Reference

STL class.

Classes

- class const_iterator

  STL iterator class.
- class const_reverse_iterator

  STL iterator class.
- class iterator

  STL iterator class.
- class reverse_iterator

  STL iterator class.

24.42.1 Detailed Description

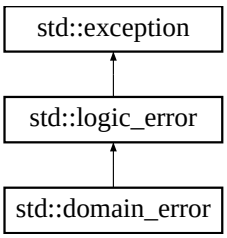
STL class.

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

24.43 std::domain_error Class Reference

STL class.

Inheritance diagram for std::domain_error:



24.43.1 Detailed Description

STL class.

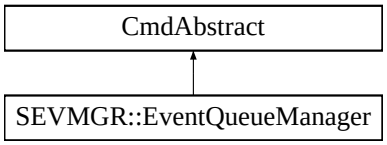
The documentation for this class was generated from the following file:

24.44 SEVMGR::EventQueueManager Class Reference

Utility class for Demand and DemandStream objects.

`#include <sevmgr/command/EventQueueManager.hpp>`

Inheritance diagram for SEVMGR::EventQueueManager:



Friends

- class SEVMGR_Service

24.44.1 Detailed Description

Utility class for Demand and DemandStream objects.

24.44.2 Friends And Related Function Documentation

24.44.2.1 friend class SEVMGR_Service [friend]

Definition at line 25 of file EventQueueManager.hpp.

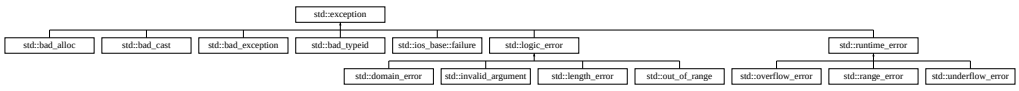
The documentation for this class was generated from the following files:

- sevmgr/command/EventQueueManager.hpp
- sevmgr/command/EventQueueManager.cpp

24.45 std::exception Class Reference

STL class.

Inheritance diagram for std::exception:



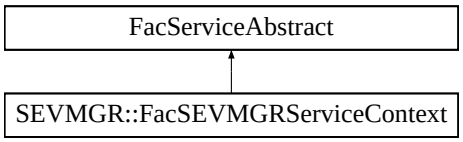
24.45.1 Detailed Description

STL class.

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

24.46 FacServiceAbstract Class Reference

Inheritance diagram for FacServiceAbstract:



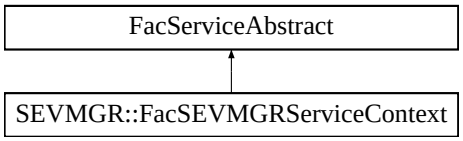
The documentation for this class was generated from the following file:

- sevmgr/factory/FacSEVMGRServiceContext.hpp

24.47 SEVMGR::FacSEVMGRServiceContext Class Reference

`#include <sevmgr/factory/FacSEVMGRServiceContext.hpp>`

Inheritance diagram for SEVMGR::FacSEVMGRServiceContext:



Public Member Functions

- ~FacSEVMGRServiceContext ()
- SEVMGR_ServiceContext & create ()

Static Public Member Functions

- static FacSEVMGRServiceContext & instance ()

Protected Member Functions

- FacSEVMGRServiceContext ()

24.47.1 Detailed Description

Factory for Bucket.

24.47.2 Constructor & Destructor Documentation

24.47.2.1 SEVMGR::FacSEVMGRServiceContext::~FacSEVMGRServiceContext ( )

Destructor.

The Destruction put the _instance to NULL in order to be clean for the next FacSEVMGRServiceContext::instance().

Definition at line 17 of file FacSEVMGRServiceContext.cpp.

24.47.2.2 SEVMGR::FacSEVMGRServiceContext::FacSEVMGRServiceContext ( ) [inline, protected]

Default Constructor.

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

Definition at line 42 of file FacSEVMGRServiceContext.hpp.

Referenced by instance().

24.47.3 Member Function Documentation

24.47.3.1 FacSEVMGRServiceContext & SEVMGR::FacSEVMGRServiceContext::instance ( ) [static]

Provide the unique instance.

The singleton is instantiated when first used

Returns

FacSEVMGRServiceContext&

Definition at line 22 of file FacSEVMGRServiceContext.cpp.

References FacSEVMGRServiceContext().

24.47.3.2 SEVMGR_ServiceContext & SEVMGR::FacSEVMGRServiceContext::create ( )

Create a new SEVMGR_ServiceContext object.

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

Returns

SEVMGR_ServiceContext& The newly created object.

Definition at line 34 of file FacSEVMGRServiceContext.cpp.

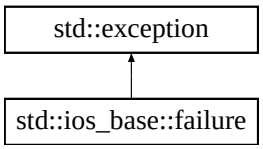
The documentation for this class was generated from the following files:

- sevmgr/factory/FacSEVMGRServiceContext.hpp
- sevmgr/factory/FacSEVMGRServiceContext.cpp

24.48 std::ios_base::failure Class Reference

STL class.

Inheritance diagram for std::ios_base::failure:



24.48.1 Detailed Description

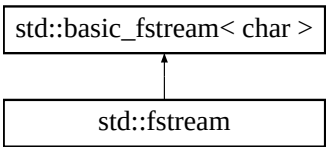
STL class.

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

24.49 std::fstream Class Reference

STL class.

Inheritance diagram for std::fstream:



24.49.1 Detailed Description

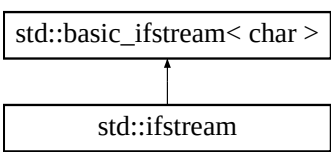
STL class.

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

24.50 std::ifstream Class Reference

STL class.

Inheritance diagram for std::ifstream:



24.50.1 Detailed Description

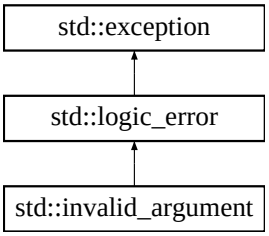
STL class.

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

24.51 std::invalid_argument Class Reference

STL class.

Inheritance diagram for std::invalid_argument:



24.51.1 Detailed Description

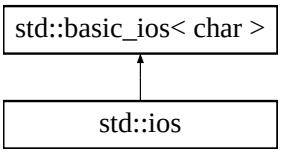
STL class.

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

24.52 std::ios Class Reference

STL class.

Inheritance diagram for std::ios:



24.52.1 Detailed Description

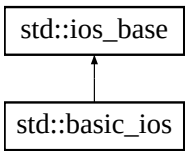
STL class.

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

24.53 std::ios_base Class Reference

STL class.

Inheritance diagram for std::ios_base:



Classes

- class failure

  STL class.

24.53.1 Detailed Description

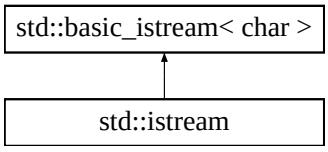
STL class.

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

24.54 std::istream Class Reference

STL class.

Inheritance diagram for std::istream:



24.54.1 Detailed Description

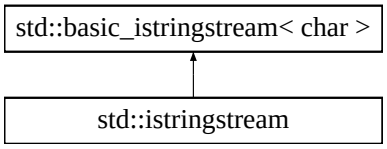
STL class.

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

24.55 std::istringstream Class Reference

STL class.

Inheritance diagram for std::istringstream:



24.55.1 Detailed Description

STL class.

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

24.56 std::list::iterator Class Reference

STL iterator class.

24.56.1 Detailed Description

STL iterator class.

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

24.57 std::deque::iterator Class Reference

STL iterator class.

24.57.1 Detailed Description

STL iterator class.

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

24.58 std::multimap::iterator Class Reference

STL iterator class.

24.58.1 Detailed Description

STL iterator class.

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

24.59 std::set::iterator Class Reference

STL iterator class.

24.59.1 Detailed Description

STL iterator class.

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

24.60 std::multiset::iterator Class Reference

STL iterator class.

24.60.1 Detailed Description

STL iterator class.

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

24.61 std::basic_string::iterator Class Reference

STL iterator class.

24.61.1 Detailed Description

STL iterator class.

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

24.62 std::vector::iterator Class Reference

STL iterator class.

24.62.1 Detailed Description

STL iterator class.

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

24.63 std::string::iterator Class Reference

STL iterator class.

24.63.1 Detailed Description

STL iterator class.

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

24.64 std::map::iterator Class Reference

STL iterator class.

24.64.1 Detailed Description

STL iterator class.

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

24.65 std::wstring::iterator Class Reference

STL iterator class.

24.65.1 Detailed Description

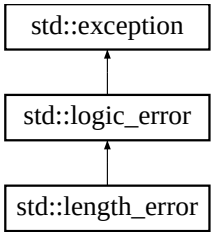
STL iterator class.

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

24.66 std::length_error Class Reference

STL class.

Inheritance diagram for std::length_error:



24.66.1 Detailed Description

STL class.

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

24.67 std::list Class Reference

STL class.

Classes

- class const_iterator

  STL iterator class.
- class const_reverse_iterator

  STL iterator class.
- class iterator

  STL iterator class.
- class reverse_iterator

  STL iterator class.

24.67.1 Detailed Description

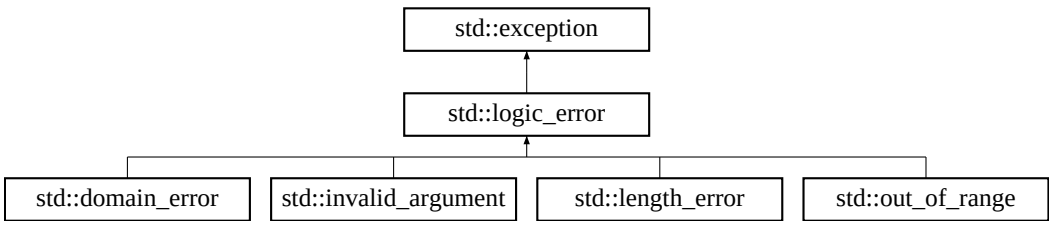
STL class.

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

24.68 std::logic_error Class Reference

STL class.

Inheritance diagram for std::logic_error:



24.68.1 Detailed Description

STL class.

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

24.69 std::map Class Reference

STL class.

Classes

- class const_iterator

  STL iterator class.
- class const_reverse_iterator

  STL iterator class.
- class iterator

  STL iterator class.
- class reverse_iterator

  STL iterator class.

24.69.1 Detailed Description

STL class.

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

24.70 std::multimap Class Reference

STL class.

Classes

- class const_iterator

  STL iterator class.
- class const_reverse_iterator

  STL iterator class.
- class iterator

  STL iterator class.
- class reverse_iterator

  STL iterator class.

24.70.1 Detailed Description

STL class.

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

24.71 std::multiset Class Reference

STL class.

Classes

- class const_iterator

  STL iterator class.
- class const_reverse_iterator

  STL iterator class.
- class iterator

  STL iterator class.
- class reverse_iterator

  STL iterator class.

24.71.1 Detailed Description

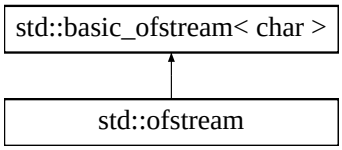
STL class.

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

24.72 std::ofstream Class Reference

STL class.

Inheritance diagram for std::ofstream:



24.72.1 Detailed Description

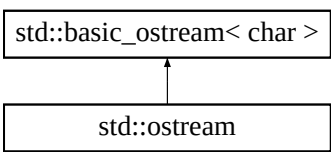
STL class.

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

24.73 std::ostream Class Reference

STL class.

Inheritance diagram for std::ostream:



24.73.1 Detailed Description

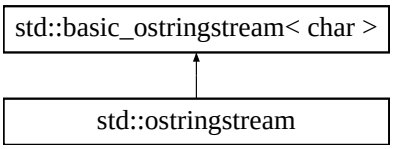
STL class.

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

24.74 std::ostringstream Class Reference

STL class.

Inheritance diagram for std::ostringstream:



24.74.1 Detailed Description

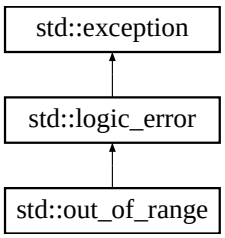
STL class.

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

24.75 std::out_of_range Class Reference

STL class.

Inheritance diagram for std::out_of_range:



24.75.1 Detailed Description

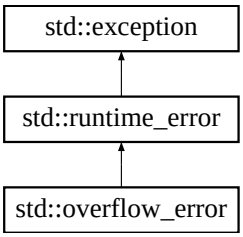
STL class.

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

24.76 std::overflow_error Class Reference

STL class.

Inheritance diagram for std::overflow_error:



24.76.1 Detailed Description

STL class.

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

24.77 std::priority_queue Class Reference

STL class.

24.77.1 Detailed Description

STL class.

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

24.78 SEVMGR::PYEventQueueManager Struct Reference

Public Member Functions

- std::string sevmgr ()
- PYEventQueueManager ()
- PYEventQueueManager (const PYEventQueueManager &iPYEventQueueManager)
- ~PYEventQueueManager ()
- bool init (const std::string &iLogFilepath, const std::string &iDBUser, const std::string &iDBPasswd, const std::string &iDBHost, const std::string &iDBPort, const std::string &iDBDBName)

24.78.1 Constructor & Destructor Documentation

24.78.1.1 SEVMGR::PYEventQueueManager::PYEventQueueManager ( ) [inline]

Default constructor.

Definition at line 76 of file pysevmgr.cpp.

24.78.1.2 SEVMGR::PYEventQueueManager::PYEventQueueManager (const PYEventQueueManager & *iPYEventQueueManager*) [inline]

Default copy constructor.

Definition at line 80 of file pysevmgr.cpp.

24.78.1.3 SEVMGR::PYEventQueueManager::~PYEventQueueManager ( ) [inline]

Default constructor.

Definition at line 86 of file pysevmgr.cpp.

24.78.2 Member Function Documentation

24.78.2.1 std::string SEVMGR::PYEventQueueManager::sevmgr ( ) [inline]

Wrapper around the travel demand generation use case.

Definition at line 25 of file pysevmgr.cpp.

References SEVMGR::SEVMGR_Service::buildSampleBom().

Referenced by BOOST_PYTHON_MODULE().

24.78.2.2 bool SEVMGR::PYEventQueueManager::init (const std::string & *iLogFilepath,* const std::string & *iDBUser,* const std::string & *iDBPasswd,* const std::string & *iDBHost,* const std::string & *iDBPort,* const std::string & *iDBDBName*) [inline]

Wrapper around the search use case.

Definition at line 92 of file pysevmgr.cpp.

Referenced by BOOST_PYTHON_MODULE().

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

- sevmgr/python/pysevmgr.cpp

24.79 std::queue Class Reference

STL class.

24.79.1 Detailed Description

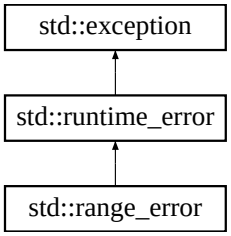
STL class.

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

24.80 std::range_error Class Reference

STL class.

Inheritance diagram for std::range_error:



24.80.1 Detailed Description

STL class.

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

24.81 std::vector::reverse_iterator Class Reference

STL iterator class.

24.81.1 Detailed Description

STL iterator class.

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

24.82 std::list::reverse_iterator Class Reference

STL iterator class.

24.82.1 Detailed Description

STL iterator class.

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

24.83 std::map::reverse_iterator Class Reference

STL iterator class.

24.83.1 Detailed Description

STL iterator class.

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

24.84 std::deque::reverse_iterator Class Reference

STL iterator class.

24.84.1 Detailed Description

STL iterator class.

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

24.85 std::wstring::reverse_iterator Class Reference

STL iterator class.

24.85.1 Detailed Description

STL iterator class.

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

24.86 std::basic_string::reverse_iterator Class Reference

STL iterator class.

24.86.1 Detailed Description

STL iterator class.

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

24.87 std::multimap::reverse_iterator Class Reference

STL iterator class.

24.87.1 Detailed Description

STL iterator class.

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

24.88 std::set::reverse_iterator Class Reference

STL iterator class.

24.88.1 Detailed Description

STL iterator class.

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

24.89 std::multiset::reverse_iterator Class Reference

STL iterator class.

24.89.1 Detailed Description

STL iterator class.

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

24.90 std::string::reverse_iterator Class Reference

STL iterator class.

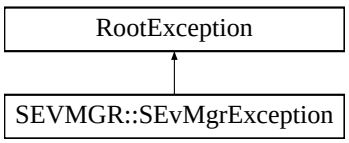
24.90.1 Detailed Description

STL iterator class.

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

24.91 RootException Class Reference

Inheritance diagram for RootException:



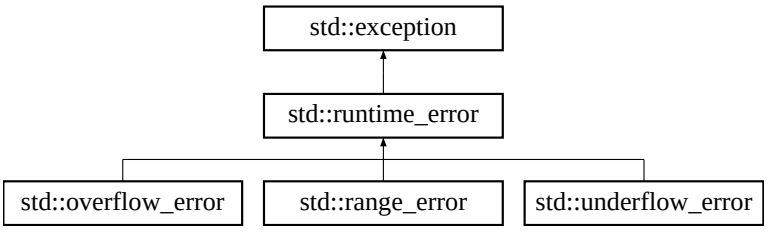
The documentation for this class was generated from the following file:

- sevmgr/SEVMGR_Exceptions.hpp

24.92 std::runtime_error Class Reference

STL class.

Inheritance diagram for std::runtime_error:



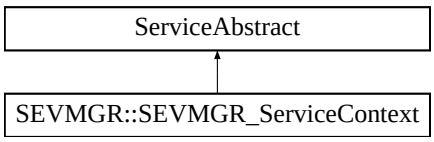
24.92.1 Detailed Description

STL class.

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

24.93 ServiceAbstract Class Reference

Inheritance diagram for ServiceAbstract:



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

- sevmgr/service/SEVMGR_ServiceContext.hpp

24.94 std::set Class Reference

STL class.

Classes

- class const_iterator

  STL iterator class.
- class const_reverse_iterator

  STL iterator class.
- class iterator

  STL iterator class.
- class reverse_iterator

  STL iterator class.

24.94.1 Detailed Description

STL class.

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

24.95 SEVMGR::SEVMGR_Service Class Reference

class holding the services related to Travel Demand Generation.

`#include <sevmgr/SEVMGR_Service.hpp>`

Public Member Functions

- SEVMGR_Service (const stdair::BasLogParams &, const stdair::BasDBParams &)

  Constructor.
- SEVMGR_Service (const stdair::BasLogParams &)
- SEVMGR_Service (stdair::STDAIR_ServicePtr_T)
- ~SEVMGR_Service ()
- void buildSampleBom ()
- stdair::ProgressStatusSet popEvent (stdair::EventStruct &) const
- bool isQueueDone () const
- void reset () const
- std::string csvDisplay () const

24.95.1 Detailed Description

class holding the services related to Travel Demand Generation.

24.95.2 Constructor & Destructor Documentation

24.95.2.1 SEVMGR::SEVMGR_Service::SEVMGR_Service (const stdair::BasLogParams & *iLogParams,* const stdair::BasDBParams & *iDBParams*)

Constructor.

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

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

Moreover, database connection parameters are given, so that a session can be created on the corresponding database.

Parameters

const	stdair::BasLogParams& Parameters for the output log stream.
const	stdair::BasDBParams& Parameters for the database access.

Definition at line 39 of file SEVMGR_Service.cpp.

24.95.2.2 SEVMGR::SEVMGR_Service::SEVMGR_Service (const stdair::BasLogParams & *iLogParams*)

Constructor.

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

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

Parameters

const	stdair::BasLogParams& Parameters for the output log stream.

Definition at line 60 of file SEVMGR_Service.cpp.

24.95.2.3 SEVMGR::SEVMGR_Service::SEVMGR_Service (stdair::STDAIR_ServicePtr_T *ioSTDAIR_Service_ptr*)

Constructor.

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

Moreover, as no reference on any output stream is given, neither any database access parameter is given, it is assumed that the StdAir log service has already been initialised with the proper log output stream by some other methods in the calling chain (for instance, when the SEVMGR_Service is itself being initialised by another library service such as DSIM_Service).

Parameters

stdair::STDAIR_ServicePtr_T	Handler on the STDAIR_Service.

Definition at line 81 of file SEVMGR_Service.cpp.

24.95.2.4 SEVMGR::SEVMGR_Service::~SEVMGR_Service ( )

Destructor.

Definition at line 97 of file SEVMGR_Service.cpp.

24.95.3 Member Function Documentation

24.95.3.1 void SEVMGR::SEVMGR_Service::buildSampleBom ( )

Build a sample BOM tree.

Definition at line 171 of file SEVMGR_Service.cpp.

Referenced by main(), and SEVMGR::PYEventQueueManager::sevmgr().

24.95.3.2 stdair::ProgressStatusSet SEVMGR::SEVMGR_Service::popEvent (stdair::EventStruct & *ioEventStruct*) const

Pop the next coming (in time) event, and remove it from the event queue.

- The next coming (in time) event corresponds to the event having the earliest date-time stamp. In other words, it is the first/front element of the event queue.
- That (first) event/element is then removed from the event queue
- The progress status is updated for the corresponding demand stream.

Returns

stdair::EventStruct A copy of the event structure, which comes first in time from within the event queue.

Definition at line 215 of file SEVMGR_Service.cpp.

24.95.3.3 bool SEVMGR::SEVMGR_Service::isQueueDone ( ) const

States whether the event queue has reached the end.

For now, that method states whether the event queue is empty.

Definition at line 233 of file SEVMGR_Service.cpp.

24.95.3.4 void SEVMGR::SEVMGR_Service::reset ( ) const

Reset the context of the demand streams for another demand generation without having to reparse the demand input file.

Definition at line 255 of file SEVMGR_Service.cpp.

24.95.3.5 std::string SEVMGR::SEVMGR_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 191 of file SEVMGR_Service.cpp.

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

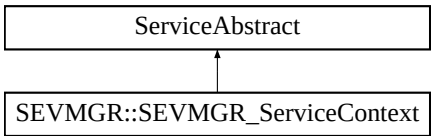
- sevmgr/SEVMGR_Service.hpp
- sevmgr/service/SEVMGR_Service.cpp

24.96 SEVMGR::SEVMGR_ServiceContext Class Reference

Class holding the context of the Sevmgr services.

`#include <sevmgr/service/SEVMGR_ServiceContext.hpp>`

Inheritance diagram for SEVMGR::SEVMGR_ServiceContext:



Friends

- class SEVMGR_Service
- class FacSEVMGRServiceContext

24.96.1 Detailed Description

Class holding the context of the Sevmgr services.

24.96.2 Friends And Related Function Documentation

24.96.2.1 friend class SEVMGR_Service [friend]

The SEVMGR_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 27 of file SEVMGR_ServiceContext.hpp.

24.96.2.2 friend class FacSEVMGRServiceContext [friend]

Definition at line 28 of file SEVMGR_ServiceContext.hpp.

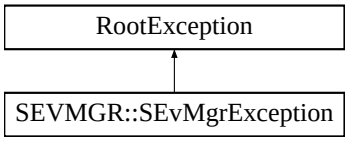
The documentation for this class was generated from the following files:

- sevmgr/service/SEVMGR_ServiceContext.hpp
- sevmgr/service/SEVMGR_ServiceContext.cpp

24.97 SEVMGR::SEvMgrException Class Reference

`#include <sevmgr/SEVMGR_Exceptions.hpp>`

Inheritance diagram for SEVMGR::SEvMgrException:



Public Member Functions

- SEvMgrException (const std::string &iWhat)

24.97.1 Detailed Description

Root exception for the Sevmgr component

24.97.2 Constructor & Destructor Documentation

24.97.2.1 SEVMGR::SEvMgrException::SEvMgrException (const std::string & *iWhat*) [inline]

Constructor.

Definition at line 23 of file SEVMGR_Exceptions.hpp.

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

- sevmgr/SEVMGR_Exceptions.hpp

24.98 swift::SKeymap Class Reference

The readline keymap wrapper.

`#include <sevmgr/ui/cmdline/SReadline.hpp>`

Public Member Functions

- SKeymap (bool PrintableBound=false)

  Creates a new keymap.
- SKeymap (Keymap Pattern)

  Creates a new keymap which is a copy of Pattern.
- ~SKeymap ()

  Frees the allocated keymap.
- void Bind (int Key, KeyCallback Callback)

  Binds the given key to a function.
- void Unbind (int Key)

  Unbinds the given key.
- SKeymap (const SKeymap &rhs)

  Copy constructor.
- SKeymap & operator= (const SKeymap &rhs)

  operator=

Friends

- class SReadline

24.98.1 Detailed Description

The readline keymap wrapper.

Attention: It is not thread safe! Supports: key binding, key unbinding

24.98.2 Constructor & Destructor Documentation

24.98.2.1 swift::SKeymap::SKeymap (bool *PrintableBound* = `false`) [inline, explicit]

Creates a new keymap.

Parameters

Printable-Bound	if true - the printable characters are bound if false - the keymap is empty

Definition at line 319 of file SReadline.hpp.

24.98.2.2 swift::SKeymap::SKeymap (Keymap *Pattern*) [inline, explicit]

Creates a new keymap which is a copy of Pattern.

Parameters

Pattern	A keymap to be copied.

Definition at line 342 of file SReadline.hpp.

24.98.2.3 swift::SKeymap::~SKeymap ( ) [inline]

Frees the allocated keymap.

Definition at line 354 of file SReadline.hpp.

24.98.2.4 swift::SKeymap::SKeymap (const SKeymap & *rhs*) [inline]

Copy constructor.

Parameters

rhs	Right hand side object of SKeymap

Definition at line 395 of file SReadline.hpp.

24.98.3 Member Function Documentation

24.98.3.1 void swift::SKeymap::Bind (int *Key*, KeyCallback *Callback*) [inline]

Binds the given key to a function.

Parameters

Key	A key to be bound
Callback	A function to be called when the Key is pressed

Definition at line 366 of file SReadline.hpp.

24.98.3.2 void swift::SKeymap::Unbind (int *Key*) [inline]

Unbinds the given key.

Parameters

Key	A key to be unbound

Definition at line 381 of file SReadline.hpp.

24.98.3.3 SKeymap& swift::SKeymap::operator= (const SKeymap & *rhs*) [inline]

operator=

Parameters

rhs	Right hand side object of SKeymap

Definition at line 407 of file SReadline.hpp.

24.98.4 Friends And Related Function Documentation

24.98.4.1 friend class SReadline [friend]

Definition at line 415 of file SReadline.hpp.

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

- sevmgr/ui/cmdline/SReadline.hpp

24.99 swift::SReadline Class Reference

The readline library wrapper.

`#include <sevmgr/ui/cmdline/SReadline.hpp>`

Public Member Functions

- SReadline (const size_t Limit=DefaultHistoryLimit)

  Constructs the object, sets the completion function.
- SReadline (const std::string &historyFileName, const size_t Limit=DefaultHistoryLimit)

  Constructs the object, sets the completion function, loads history.
- ~SReadline ()

  Saves the session history (if the file name was provided) and destroys the object.
- std::string GetLine (const std::string &Prompt)

  Gets a single line from a user.
- template<typename Container >
  std::string GetLine (const std::string &Prompt, Container &ReadTokens)

  Gets a single line from a user.
- template<typename Container >
  std::string GetLine (const std::string &Prompt, Container &ReadTokens, bool &BreakOut)

  Gets a single line from a user.
- std::string GetLine (const std::string &Prompt, bool &BreakOut)

  Gets a single line from a user.

- template<typename ContainerType >
  void GetHistory (ContainerType &Container)

  Fills the given container with the current history list.
- bool SaveHistory (std::ostream &OS)

  Saves the history to the given file stream.
- bool SaveHistory (const std::string &FileName)

  Saves the history to the given file.
- void ClearHistory ()

  Clears the history. Does not affect the file where the previous session history is saved.
- bool LoadHistory (std::istream &IS)

  Loads a history from a file stream.
- bool LoadHistory (const std::string &FileName)

  Loads a history from the given file.
- template<typename ContainerType >
  void RegisterCompletions (const ContainerType &Container)

  Allows to register custom completers.
- void SetKeymap (SKeymap &NewKeymap)

  Sets the given keymap.

24.99.1 Detailed Description

The readline library wrapper.

Attention: It is not thread safe! Supports: editing, history, custom completers

24.99.2 Constructor & Destructor Documentation

24.99.2.1 swift::SReadline::SReadline (const size_t *Limit* = `DefaultHistoryLimit`) `[inline]`

Constructs the object, sets the completion function.

Parameters

Limit	History size

Definition at line 431 of file SReadline.hpp.

24.99.2.2 swift::SReadline::SReadline (const std::string & *historyFileName*, const size_t *Limit* = `DefaultHistoryLimit`) `[inline]`

Constructs the object, sets the completion function, loads history.

Parameters

historyFile-Name	File name to load history from
Limit	History size

Definition at line 446 of file SReadline.hpp.

References LoadHistory().

24.99.2.3 swift::SReadline::~SReadline ( ) [inline]

Saves the session history (if the file name was provided) and destroys the object.

Definition at line 460 of file SReadline.hpp.

References SaveHistory().

24.99.3 Member Function Documentation

24.99.3.1 std::string swift::SReadline::GetLine (const std::string & *Prompt*) [inline]

Gets a single line from a user.

Parameters

Prompt	A printed prompt

Returns

A string which was actually inputed

Definition at line 471 of file SReadline.hpp.

Referenced by GetLine().

24.99.3.2 template<typename Container > std::string swift::SReadline::GetLine (const std::string & *Prompt,* Container & *ReadTokens*) [inline]

Gets a single line from a user.

Parameters

Prompt	A printed prompt
ReadTokens	A user inputed string splitted into tokens. The container is cleared first

Returns

A string which was actually inputed

Definition at line 485 of file SReadline.hpp.

References GetLine().

24.99.3.3 template<typename Container > std::string swift::SReadline::GetLine (const std::string & *Prompt,* Container & *ReadTokens,* bool & *BreakOut*) [inline]

Gets a single line from a user.

Parameters

Prompt	A printed prompt
BreakOut	it is set to true if the EOF found
ReadTokens	A user inputed string splitted into tokens. The container is cleared first

Returns

A string which was actually inputed

Definition at line 500 of file SReadline.hpp.

References GetLine().

24.99.3.4 std::string swift::SReadline::GetLine (const std::string & *Prompt*, bool & *BreakOut*) [inline]

Gets a single line from a user.

Parameters

Prompt	A printed prompt
BreakOut	it is set to true if the EOF found

Returns

A string which was actually inputed

Definition at line 515 of file SReadline.hpp.

24.99.3.5 template<typename ContainerType > void swift::SReadline::GetHistory (ContainerType & *Container*) [inline]

Fills the given container with the current history list.

Does not clear the given container

Definition at line 550 of file SReadline.hpp.

24.99.3.6 bool swift::SReadline::SaveHistory (std::ostream & *OS*) [inline]

Saves the history to the given file stream.

Parameters

OS	output file stream

Returns

true if success

Definition at line 562 of file SReadline.hpp.

Referenced by ~SReadline(), and SaveHistory().

24.99.3.7 bool swift::SReadline::SaveHistory (const std::string & *FileName*) [inline]

Saves the history to the given file.

Parameters

FileName	File name to save the history to

Returns

true if success

Definition at line 579 of file SReadline.hpp.

References SaveHistory().

24.99.3.8 void swift::SReadline::ClearHistory ( ) [inline]

Clears the history. Does not affect the file where the previous session history is saved.

Definition at line 592 of file SReadline.hpp.

Referenced by LoadHistory().

24.99.3.9 bool swift::SReadline::LoadHistory (std::istream & *IS*) [inline]

Loads a history from a file stream.

Parameters

IS	Input file stream

Returns

true if success

Definition at line 602 of file SReadline.hpp.

References ClearHistory().

Referenced by SReadline(), and LoadHistory().

24.99.3.10 bool swift::SReadline::LoadHistory (const std::string & *FileName*) [inline]

Loads a history from the given file.

Parameters

FileName	File name to be load from

Returns

true if success

Definition at line 627 of file SReadline.hpp.

References LoadHistory().

24.99.3.11 template<typename ContainerType > void swift::SReadline::RegisterCompletions (const ContainerType & *Container*) [inline]

Allows to register custom completers.

Supports a special keyword: file. It means to use the standard file name completer.

For example the given container elements could be as follows:

- command1 opt1
- command1 opt2 file
- command2
- command2 opt1

Each container element must describe a single possible command line. The container element must have a conversion to std::string operator.

Parameters

Container	A container which has all the user possible commands.

Definition at line 656 of file SReadline.hpp.

24.99.3.12 void swift::SReadline::SetKeymap (SKeymap & *NewKeymap*) [inline]

Sets the given keymap.

Parameters

NewKeymap	The keymap that should be used from now.

Definition at line 673 of file SReadline.hpp.

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

- sevmgr/ui/cmdline/SReadline.hpp

24.100 std::stack Class Reference

STL class.

24.100.1 Detailed Description

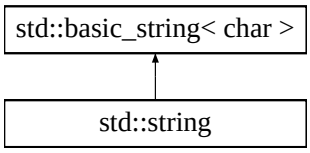
STL class.

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

24.101 std::string Class Reference

STL class.

Inheritance diagram for std::string:



Classes

- class const_iterator
  STL iterator class.
- class const_reverse_iterator
  STL iterator class.
- class iterator
  STL iterator class.
- class reverse_iterator
  STL iterator class.

24.101.1 Detailed Description

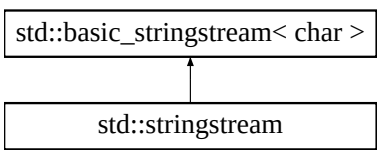
STL class.

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

24.102 std::stringstream Class Reference

STL class.

Inheritance diagram for std::stringstream:



24.102.1 Detailed Description

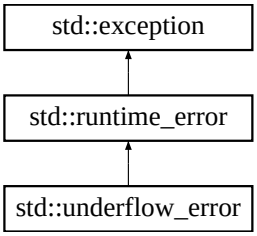
STL class.

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

24.103 std::underflow_error Class Reference

STL class.

Inheritance diagram for std::underflow_error:



24.103.1 Detailed Description

STL class.

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

24.104 std::valarray Class Reference

STL class.

24.104.1 Detailed Description

STL class.

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

24.105 std::vector Class Reference

STL class.

Classes

- class const_iterator

  STL iterator class.
- class const_reverse_iterator

STL iterator class.

- class iterator

  STL iterator class.

- class reverse_iterator

  STL iterator class.

24.105.1 Detailed Description

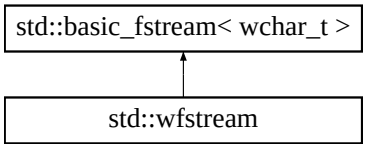
STL class.

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

24.106 std::wfstream Class Reference

STL class.

Inheritance diagram for std::wfstream:



24.106.1 Detailed Description

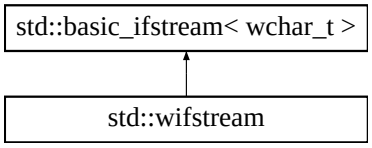
STL class.

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

24.107 std::wifstream Class Reference

STL class.

Inheritance diagram for std::wifstream:



24.107.1 Detailed Description

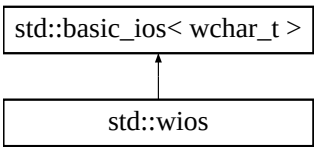
STL class.

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

24.108 std::wios Class Reference

STL class.

Inheritance diagram for std::wios:



24.108.1 Detailed Description

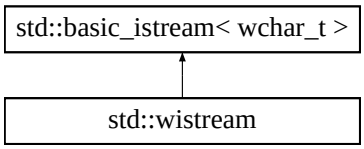
STL class.

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

24.109 std::wistream Class Reference

STL class.

Inheritance diagram for std::wistream:



24.109.1 Detailed Description

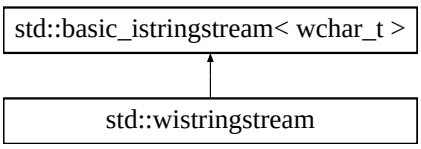
STL class.

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

24.110 std::wistringstream Class Reference

STL class.

Inheritance diagram for std::wistringstream:



24.110.1 Detailed Description

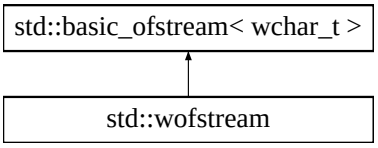
STL class.

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

24.111 std::wofstream Class Reference

STL class.

Inheritance diagram for std::wofstream:



24.111.1 Detailed Description

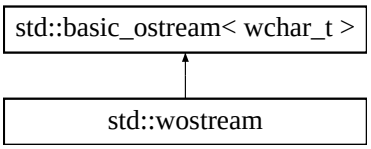
STL class.

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

24.112 std::wostream Class Reference

STL class.

Inheritance diagram for std::wostream:



24.112.1 Detailed Description

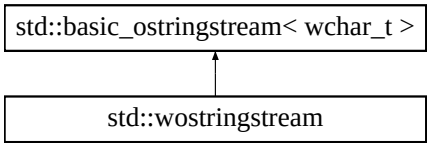
STL class.

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

24.113 std::wostringstream Class Reference

STL class.

Inheritance diagram for std::wostringstream:



24.113.1 Detailed Description

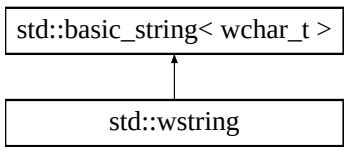
STL class.

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

24.114 std::wstring Class Reference

STL class.

Inheritance diagram for std::wstring:



Classes

- class const_iterator
  STL iterator class.
- class const_reverse_iterator
  STL iterator class.
- class iterator
  STL iterator class.
- class reverse_iterator
  STL iterator class.

24.114.1 Detailed Description

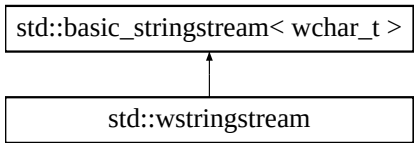
STL class.

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

24.115 std::wstringstream Class Reference

STL class.

Inheritance diagram for std::wstringstream:



24.115.1 Detailed Description

STL class.

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

25 File Documentation

25.1 doc/local/authors.doc File Reference

25.2 doc/local/codingrules.doc File Reference

25.3 doc/local/copyright.doc File Reference

25.4 doc/local/documentation.doc File Reference

25.5 doc/local/features.doc File Reference

25.6 doc/local/help_wanted.doc File Reference

25.7 doc/local/howto_release.doc File Reference

25.8 doc/local/index.doc File Reference

25.9 doc/local/installation.doc File Reference

25.10 doc/local/linking.doc File Reference

25.11 doc/local/test.doc File Reference

25.12 doc/local/users_guide.doc File Reference

25.13 doc/local/verification.doc File Reference

25.14 doc/tutorial/tutorial.doc File Reference

25.15 sevmgr/basic/BasConst.cpp File Reference

```
#include <stdair/basic/BasConst_General.hpp>     #include
<sevmgr/basic/BasConst_SEVMGR_Service.hpp> #include <sevmgr/basic/-
BasConst_EventQueueManager.hpp>
```

Namespaces

- namespace SEVMGR

25.16 BasConst.cpp

```
00001 // ////////////////////////////////////////////////////////////////////
00002 // Import section
00003 // ////////////////////////////////////////////////////////////////////
00004 // StdAir
00005 #include <stdair/basic/BasConst_General.hpp>
00006 // Sevmgr
00007 #include <sevmgr/basic/BasConst_SEVMGR_Service.hpp>
00008 #include <sevmgr/basic/BasConst_EventQueueManager.hpp>
00009
00010 namespace SEVMGR {
00011
00013   // const std::string DEFAULT_SEVMGR_SERVICE_NAME = "sevmgr";
00014
00016   // extern const std::string DEFAULT_EVENT_QUEUE_KEY;
00017
00018 }
```

25.17 sevmgr/basic/BasConst_EventQueueManager.hpp File Reference

```
#include <string>
```

Namespaces

- namespace SEVMGR

25.18 BasConst_EventQueueManager.hpp

```
00001 #ifndef __SEVMGR_BAS_BASCONST_EVENTQUEUEMANAGER_HPP
```

```
00002 #define __SEVMGR_BAS_BASCONST_EVENTQUEUEMANAGER_HPP
00003
00004 // ////////////////////////////////////////////////////////////////////
00005 // Import section
00006 // ////////////////////////////////////////////////////////////////////
00007 // STL
00008 #include <string>
00009
00010 namespace SEVMGR {
00011
00013   // extern const std::string DEFAULT_EVENT_QUEUE_KEY;
00014
00015 }
00016 #endif // __SEVMGR_BAS_BASCONST_EVENTQUEUEMANAGER_HPP
```

25.19 sevmgr/basic/BasConst_SEVMGR_Service.hpp File Reference

```
#include <string>
```

Namespaces

- namespace SEVMGR

25.20 BasConst_SEVMGR_Service.hpp

```
00001 #ifndef __SEVMGR_BAS_BASCONST_SEVMGR_SERVICE_HPP
00002 #define __SEVMGR_BAS_BASCONST_SEVMGR_SERVICE_HPP
00003
00004 // ////////////////////////////////////////////////////////////////////
00005 // Import section
00006 // ////////////////////////////////////////////////////////////////////
00007 #include <string>
00008
00009 namespace SEVMGR {
00010
00012   // extern const std::string DEFAULT_SEVMGR_SERVICE_NAME;
00013
00014 }
00015 #endif // __SEVMGR_BAS_BASCONST_SEVMGR_SERVICE_HPP
```

25.21 sevmgr/basic/BasParserTypes.hpp File Reference

```
#include <string> #include <boost/spirit/home/classic/core.-
hpp>  #include <boost/spirit/home/classic/utility/loops.-
hpp>  #include <boost/spirit/home/classic/utility/chset.-
hpp> #include <boost/spirit/home/classic/utility/confix.-
hpp>  #include <boost/spirit/home/classic/iterator/file_-
iterator.hpp>
```

Namespaces

- namespace SEVMGR

Typedefs

- typedef char SEVMGR::char_t
- typedef boost::spirit::classic::file_iterator < char_t > SEVMGR::iterator_t
- typedef boost::spirit::classic::scanner < iterator_t > SEVMGR::scanner_t
- typedef boost::spirit::classic::rule < scanner_t > SEVMGR::rule_t
- typedef boost::spirit::classic::int_parser < unsigned int, 10, 1, 1 > SEVMGR::int1_p_t
- typedef boost::spirit::classic::uint_parser < unsigned int, 10, 2, 2 > SEVMGR::uint2_p_t
- typedef boost::spirit::classic::uint_parser < unsigned int, 10, 1, 2 > SEVMGR::uint1_2_p_t
- typedef boost::spirit::classic::uint_parser < unsigned int, 10, 1, 3 > SEVMGR::uint1_3_p_t
- typedef boost::spirit::classic::uint_parser < unsigned int, 10, 4, 4 > SEVMGR::uint4_p_t
- typedef boost::spirit::classic::uint_parser < unsigned int, 10, 1, 4 > SEVMGR::uint1_4_p_t
- typedef boost::spirit::classic::chset < char_t > SEVMGR::chset_t
- typedef boost::spirit::classic::impl::loop_traits < chset_t, unsigned int, unsigned int >::type SEVMGR::repeat_p_t
- typedef boost::spirit::classic::bounded < uint2_p_t, unsigned int > SEVMGR::bounded2_p_t
- typedef boost::spirit::classic::bounded < uint1_2_p_t, unsigned int > SEVMGR::bounded1_2_p_t
- typedef boost::spirit::classic::bounded < uint1_3_p_t, unsigned int > SEVMGR::bounded1_3_p_t
- typedef boost::spirit::classic::bounded < uint4_p_t, unsigned int > SEVMGR::bounded4_p_t
- typedef boost::spirit::classic::bounded < uint1_4_p_t, unsigned int > SEVMGR::bounded1_4_p_t

25.22 BasParserTypes.hpp

```
00001 #ifndef __SEVMGR_BAS_BASCOMPARSERTYPES_HPP
00002 #define __SEVMGR_BAS_BASCOMPARSERTYPES_HPP
00003
00004 // ////////////////////////////////////////////////////////////////////
00005 // Import section
00006 // ////////////////////////////////////////////////////////////////////
00007 // STL
00008 #include <string>
00009 // Boost
00010 //#define BOOST_SPIRIT_DEBUG
00011 #include <boost/spirit/home/classic/core.hpp>
00012 //#include <boost/spirit/home/classic/attribute.hpp>
00013 //#include <boost/spirit/home/classic/utility/functor_parser.hpp>
00014 #include <boost/spirit/home/classic/utility/loops.hpp>
00015 #include <boost/spirit/home/classic/utility/chset.hpp>
00016 #include <boost/spirit/home/classic/utility/confix.hpp>
00017 #include <boost/spirit/home/classic/iterator/file_iterator.hpp>
00018 //#include <boost/spirit/home/classic/actor/push_back_actor.hpp>
00019 //#include <boost/spirit/home/classic/actor/assign_actor.hpp>
00020
00021 namespace SEVMGR {
```

```
00022
00023   // ////////////////////////////////////////////////////////////////////////
00024   //
00025   //  Definition of Basic Types
00026   //
00027   // ////////////////////////////////////////////////////////////////////////
00028   // For a file, the parsing unit is the character (char). For a string,
00029   // it is a "char const *".
00030   // typedef char const* iterator_t;
00031   typedef char char_t;
00032
00033   // The types of iterator, scanner and rule are then derived from
00034   // the parsing unit.
00035   typedef boost::spirit::classic::file_iterator<char_t> iterator_t;
00036   typedef boost::spirit::classic::scanner<iterator_t> scanner_t;
00037   typedef boost::spirit::classic::rule<scanner_t> rule_t;
00038
00039   // ////////////////////////////////////////////////////////////////////////
00040   //
00041   //  Parser related types
00042   //
00043   // ////////////////////////////////////////////////////////////////////////
00045   typedef boost::spirit::classic::int_parser<unsigned int, 10, 1, 1> int1_p_t;
00046
00048   typedef boost::spirit::classic::uint_parser<unsigned int, 10, 2, 2> uint2_p_t
      ;
00049
00051   typedef boost::spirit::classic::uint_parser<unsigned int, 10, 1, 2>
      uint1_2_p_t;
00052
00054   typedef boost::spirit::classic::uint_parser<unsigned int, 10, 1, 3>
      uint1_3_p_t;
00055
00057   typedef boost::spirit::classic::uint_parser<unsigned int, 10, 4, 4> uint4_p_t
      ;
00058
00060   typedef boost::spirit::classic::uint_parser<unsigned int, 10, 1, 4>
      uint1_4_p_t;
00061
00063   typedef boost::spirit::classic::chset<char_t> chset_t;
00064
00067   typedef boost::spirit::classic::impl::loop_traits<chset_t,
00068                                                     unsigned int,
00069                                                     unsigned int>::type repeat_p_t;
00070
00072   typedef boost::spirit::classic::bounded<uint2_p_t, unsigned int> bounded2_p_t
      ;
00073   typedef boost::spirit::classic::bounded<uint1_2_p_t, unsigned int>
      bounded1_2_p_t;
00074   typedef boost::spirit::classic::bounded<uint1_3_p_t, unsigned int>
      bounded1_3_p_t;
00075   typedef boost::spirit::classic::bounded<uint4_p_t, unsigned int> bounded4_p_t
      ;
00076   typedef boost::spirit::classic::bounded<uint1_4_p_t, unsigned int>
      bounded1_4_p_t;
00077 }
00078 #endif // __SEVMGR_BAS_BASCOMPARSERTYPES_HPP
```

25.23 sevmgr/batches/sevmgr_demo.cpp File Reference

```
#include <cassert> #include <sstream> #include <fstream> ×
#include <vector>   #include <list>   #include <string> ×
#include <boost/program_options.hpp> #include <stdair/stdair-
_basic_types.hpp>  #include <stdair/basic/ProgressStatus-
Set.hpp>  #include <stdair/bom/EventStruct.hpp>  #include
<stdair/bom/EventQueue.hpp>     #include <stdair/bom/Bom-
Display.hpp> #include <stdair/service/Logger.hpp> #include
```

<sevmgr/SEVMGR_Service.hpp> #include <sevmgr/config/sevmgr-paths.-hpp>

Functions

- const stdair::Filename_T K_SEVMGR_DEFAULT_LOG_FILENAME ("sevmgr_-demo.log")
- int readConfiguration (int argc, char ∗argv[], stdair::Filename_T &ioLogFilename)
- int main (int argc, char ∗argv[])

Variables

- const int K_SEVMGR_EARLY_RETURN_STATUS = 99

25.23.1 Function Documentation

25.23.1.1 const stdair::Filename_T K_SEVMGR_DEFAULT_LOG_FILENAME ("sevmgr_demo.log")

Default name and location for the log file.

Referenced by readConfiguration().

25.23.1.2 int readConfiguration (int *argc,* char ∗ *argv[],* stdair::Filename_T & *ioLogFilename*)

Read and parse the command line options.

Definition at line 35 of file sevmgr_demo.cpp.

References K_SEVMGR_DEFAULT_LOG_FILENAME(), K_SEVMGR_EARLY_RET-URN_STATUS, PACKAGE_NAME, PACKAGE_VERSION, and PREFIXDIR.

Referenced by main().

25.23.1.3 int main (int *argc,* char ∗ *argv[]*)

Definition at line 109 of file sevmgr_demo.cpp.

References readConfiguration(), K_SEVMGR_EARLY_RETURN_STATUS, and SEV-MGR::SEVMGR_Service::buildSampleBom().

25.23.2 Variable Documentation

25.23.2.1 const int K_SEVMGR_EARLY_RETURN_STATUS = 99

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

Definition at line 30 of file sevmgr_demo.cpp.

Referenced by readConfiguration(), and main().

25.24 sevmgr_demo.cpp

```
00001 // ////////////////////////////////////////////////////////////////////
00002 // Import section
00003 // ////////////////////////////////////////////////////////////////////
00004 // STL
00005 #include <cassert>
00006 #include <sstream>
00007 #include <fstream>
00008 #include <vector>
00009 #include <list>
00010 #include <string>
00011 //  //// Boost (Extended STL) ////
00012 // Boost Program Options
00013 #include <boost/program_options.hpp>
00014 // StdAir
00015 #include <stdair/stdair_basic_types.hpp>
00016 #include <stdair/basic/ProgressStatusSet.hpp>
00017 #include <stdair/bom/EventStruct.hpp>
00018 #include <stdair/bom/EventQueue.hpp>
00019 #include <stdair/bom/BomDisplay.hpp>
00020 #include <stdair/service/Logger.hpp>
00021 // SEvMgr
00022 #include <sevmgr/SEVMGR_Service.hpp>
00023 #include <sevmgr/config/sevmgr-paths.hpp>
00024
00025 // //////// Constants //////
00027 const stdair::Filename_T K_SEVMGR_DEFAULT_LOG_FILENAME ("sevmgr_demo.log");
00028
00030 const int K_SEVMGR_EARLY_RETURN_STATUS = 99;
00031
00032
00033 // ///////// Parsing of Options & Configuration /////////
00035 int readConfiguration (int argc, char* argv[],
00036                        stdair::Filename_T& ioLogFilename) {
00037
00038   // Declare a group of options that will be allowed only on command line
00039   boost::program_options::options_description generic ("Generic options");
00040   generic.add_options()
00041     ("prefix", "print installation prefix")
00042     ("version,v", "print version string")
00043     ("help,h", "produce help message");
00044
00045   // Declare a group of options that will be allowed both on command
00046   // line and in config file
00047   boost::program_options::options_description config ("Configuration");
00048   config.add_options()
00049     ("log,l",
00050      boost::program_options::value< std::string >(&ioLogFilename)->
      default_value(K_SEVMGR_DEFAULT_LOG_FILENAME),
00051      "Filepath for the logs")
00052     ;
00053
00054   // Hidden options, will be allowed both on command line and
00055   // in config file, but will not be shown to the user.
00056   boost::program_options::options_description hidden ("Hidden options");
00057   hidden.add_options()
00058     ("copyright",
00059      boost::program_options::value< std::vector<std::string> >(),
00060      "Show the copyright (license)");
00061
00062   boost::program_options::options_description cmdline_options;
00063   cmdline_options.add(generic).add(config).add(hidden);
00064
00065   boost::program_options::options_description config_file_options;
00066   config_file_options.add(config).add(hidden);
00067
00068   boost::program_options::options_description visible ("Allowed options");
00069   visible.add(generic).add(config);
00070
00071   boost::program_options::positional_options_description p;
00072   p.add ("copyright", -1);
00073
```

```
00074   boost::program_options::variables_map vm;
00075   boost::program_options::
00076     store (boost::program_options::command_line_parser (argc, argv).
00077            options (cmdline_options).positional(p).run(), vm);
00078
00079   std::ifstream ifs ("sevmgr.cfg");
00080   boost::program_options::store (parse_config_file (ifs, config_file_options),
00081                                  vm);
00082   boost::program_options::notify (vm);
00083
00084   if (vm.count ("help")) {
00085     std::cout << visible << std::endl;
00086     return K_SEVMGR_EARLY_RETURN_STATUS;
00087   }
00088
00089   if (vm.count ("version")) {
00090     std::cout << PACKAGE_NAME << ", version " << PACKAGE_VERSION << std::endl;
00091     return K_SEVMGR_EARLY_RETURN_STATUS;
00092   }
00093
00094   if (vm.count ("prefix")) {
00095     std::cout << "Installation prefix: " << PREFIXDIR << std::endl;
00096     return K_SEVMGR_EARLY_RETURN_STATUS;
00097   }
00098
00099   if (vm.count ("log")) {
00100     ioLogFilename = vm["log"].as< std::string >();
00101     std::cout << "Log filename is: " << ioLogFilename << std::endl;
00102   }
00103
00104   return 0;
00105 }
00106
00107
00108 // /////////////// M A I N /////////////////
00109 int main (int argc, char* argv[]) {
00110
00111   // Output log File
00112   stdair::Filename_T lLogFilename;
00113
00114   // Call the command-line option parser
00115   const int lOptionParserStatus = readConfiguration (argc, argv, lLogFilename);
00116
00117   if (lOptionParserStatus == K_SEVMGR_EARLY_RETURN_STATUS) {
00118     return 0;
00119   }
00120
00121   // Set the log parameters
00122   std::ofstream logOutputFile;
00123   // Open and clean the log outputfile
00124   logOutputFile.open (lLogFilename.c_str());
00125   logOutputFile.clear();
00126
00127   // Set up the log parameters
00128   const stdair::BasLogParams lLogParams (stdair::LOG::DEBUG, logOutputFile);
00129
00133   SEVMGR::SEVMGR_Service sevmgrService (lLogParams);
00134
00135   // Build a sample BOM tree (empty for now)
00136   sevmgrService.buildSampleBom();
00137
00138   // Close the Log outputFile
00139   logOutputFile.close();
00140
00141   /*
00142     Note: as that program is not intended to be run on a server in
00143     production, it is better not to catch the exceptions. When it
00144     happens (that an exception is throwned), that way we get the
00145     call stack.
00146   */
00147
00148   return 0;
00149 }
```

25.25 sevmgr/command/EventQueueManager.cpp File Reference

`#include <cassert> #include <stdair/basic/ProgressStatusSet.hpp> #include <stdair/bom/BomManager.hpp> #include <stdair/bom/EventStruct.hpp> #include <stdair/bom/EventQueue.hpp> #include <stdair/service/Logger.hpp> #include <sevmgr/command/EventQueueManager.hpp>`

Namespaces

- namespace SEVMGR

25.26 EventQueueManager.cpp

```
00001 // ////////////////////////////////////////////////////////////////////
00002 // Import section
00003 // ////////////////////////////////////////////////////////////////////
00004 // STL
00005 #include <cassert>
00006 // StdAir
00007 #include <stdair/basic/ProgressStatusSet.hpp>
00008 #include <stdair/bom/BomManager.hpp>
00009 #include <stdair/bom/EventStruct.hpp>
00010 #include <stdair/bom/EventQueue.hpp>
00011 #include <stdair/service/Logger.hpp>
00012 // SEvMgr
00013 #include <sevmgr/command/EventQueueManager.hpp>
00014
00015 namespace SEVMGR {
00016
00017   // ////////////////////////////////////////////////////////////////////
00018   void EventQueueManager::
00019   buildSampleBom (stdair::EventQueue& ioEventQueue) {
00020   }
00021
00022   // ////////////////////////////////////////////////////////////////////
00023   void EventQueueManager::reset (stdair::EventQueue& ioEventQueue) {
00024
00028     ioEventQueue.reset();
00029   }
00030
00031 }
```

25.27 sevmgr/command/EventQueueManager.hpp File Reference

`#include <stdair/stdair_basic_types.hpp> #include <stdair/command/CmdAbstract.hpp> #include <sevmgr/SEVMGR_Types.hpp>`

Classes

- class SEVMGR::EventQueueManager

  Utility class for Demand and DemandStream objects.

Namespaces

- namespace stdair
- namespace SEVMGR

25.28 EventQueueManager.hpp

```
00001 #ifndef __SEVMGR_CMD_EVENTQUEUEMANAGER_HPP
00002 #define __SEVMGR_CMD_EVENTQUEUEMANAGER_HPP
00003
00004 // //////////////////////////////////////////////////////////////////
00005 // Import section
00006 // //////////////////////////////////////////////////////////////////
00007 // StdAir
00008 #include <stdair/stdair_basic_types.hpp>
00009 #include <stdair/command/CmdAbstract.hpp>
00010 // SEvMgr
00011 #include <sevmgr/SEVMGR_Types.hpp>
00012
00013 // Forward declarations
00014 namespace stdair {
00015   class EventQueue;
00016   struct ProgressStatusSet;
00017 }
00018
00019 namespace SEVMGR {
00020
00024   class EventQueueManager : public stdair::CmdAbstract {
00025     friend class SEVMGR_Service;
00026
00027   private:
00028     // //////// Business methodes //////////
00032     static void buildSampleBom (stdair::EventQueue&);
00033
00039     static void reset (stdair::EventQueue&);
00040   };
00041
00042 }
00043 #endif // __SEVMGR_CMD_EVENTQUEUEMANAGER_HPP
```

25.29 sevmgr/config/sevmgr-paths.hpp File Reference

Defines

- #define PACKAGE "sevmgr"
- #define PACKAGE_NAME "SEVMGR"
- #define PACKAGE_VERSION "0.2.0"
- #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/sevmgr-0.2.0"

- #define MANDIR "/usr/share/man"
- #define INFODIR "/usr/share/info"
- #define HTMLDIR "/usr/share/doc/sevmgr-0.2.0/html"
- #define PDFDIR "/usr/share/doc/sevmgr-0.2.0/html"
- #define STDAIR_SAMPLE_DIR "/usr/share/stdair/samples"

25.29.1 Define Documentation

25.29.1.1 #define PACKAGE "sevmgr"

Definition at line 4 of file sevmgr-paths.hpp.

25.29.1.2 #define PACKAGE_NAME "SEVMGR"

Definition at line 5 of file sevmgr-paths.hpp.

Referenced by readConfiguration().

25.29.1.3 #define PACKAGE_VERSION "0.2.0"

Definition at line 6 of file sevmgr-paths.hpp.

Referenced by readConfiguration().

25.29.1.4 #define PREFIXDIR "/usr"

Definition at line 7 of file sevmgr-paths.hpp.

Referenced by readConfiguration().

25.29.1.5 #define EXEC_PREFIX "/usr"

Definition at line 8 of file sevmgr-paths.hpp.

25.29.1.6 #define BINDIR "/usr/bin"

Definition at line 9 of file sevmgr-paths.hpp.

25.29.1.7 #define LIBDIR "/usr/lib"

Definition at line 10 of file sevmgr-paths.hpp.

25.29.1.8 #define LIBEXECDIR "/usr/libexec"

Definition at line 11 of file sevmgr-paths.hpp.

25.29.1.9 #define SBINDIR "/usr/sbin"

Definition at line 12 of file sevmgr-paths.hpp.

25.29.1.10 #define SYSCONFDIR "/usr/etc"

Definition at line 13 of file sevmgr-paths.hpp.

25.29.1.11 #define INCLUDEDIR "/usr/include"

Definition at line 14 of file sevmgr-paths.hpp.

25.29.1.12 #define DATAROOTDIR "/usr/share"

Definition at line 15 of file sevmgr-paths.hpp.

25.29.1.13 #define DATADIR "/usr/share"

Definition at line 16 of file sevmgr-paths.hpp.

25.29.1.14 #define DOCDIR "/usr/share/doc/sevmgr-0.2.0"

Definition at line 17 of file sevmgr-paths.hpp.

25.29.1.15 #define MANDIR "/usr/share/man"

Definition at line 18 of file sevmgr-paths.hpp.

25.29.1.16 #define INFODIR "/usr/share/info"

Definition at line 19 of file sevmgr-paths.hpp.

25.29.1.17 #define HTMLDIR "/usr/share/doc/sevmgr-0.2.0/html"

Definition at line 20 of file sevmgr-paths.hpp.

25.29.1.18 #define PDFDIR "/usr/share/doc/sevmgr-0.2.0/html"

Definition at line 21 of file sevmgr-paths.hpp.

25.29.1.19 #define STDAIR_SAMPLE_DIR "/usr/share/stdair/samples"

Definition at line 22 of file sevmgr-paths.hpp.

25.30 sevmgr-paths.hpp

```
00001 #ifndef __SEVMGR_PATHS_HPP__
00002 #define __SEVMGR_PATHS_HPP__
00003
00004 #define PACKAGE "sevmgr"
00005 #define PACKAGE_NAME "SEVMGR"
00006 #define PACKAGE_VERSION "0.2.0"
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/sevmgr-0.2.0"
00018 #define MANDIR "/usr/share/man"
00019 #define INFODIR "/usr/share/info"
00020 #define HTMLDIR "/usr/share/doc/sevmgr-0.2.0/html"
```

```
00021 #define PDFDIR "/usr/share/doc/sevmgr-0.2.0/html"
00022 #define STDAIR_SAMPLE_DIR "/usr/share/stdair/samples"
00023
00024 #endif // __SEVMGR_PATHS_HPP__
```

25.31 sevmgr/config/sevmgr-paths.hpp.in File Reference

Defines

- #define __SEVMGR_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@"

25.31.1 Define Documentation

25.31.1.1 #define __SEVMGR_PATHS_HPP__

Definition at line 2 of file sevmgr-paths.hpp.in.

25.31.1.2 #define PACKAGE "@PACKAGE@"

Definition at line 4 of file sevmgr-paths.hpp.in.

25.31.1.3 #define PACKAGE_NAME "@PACKAGE_NAME@"

Definition at line 5 of file sevmgr-paths.hpp.in.

25.31.1.4 #define PACKAGE_VERSION "@PACKAGE_VERSION@"

Definition at line 6 of file sevmgr-paths.hpp.in.

25.31.1.5 #define PREFIXDIR "@prefix@"

Definition at line 7 of file sevmgr-paths.hpp.in.

25.31.1.6 #define EXEC_PREFIX "@exec_prefix@"

Definition at line 8 of file sevmgr-paths.hpp.in.

25.31.1.7 #define BINDIR "@bindir@"

Definition at line 9 of file sevmgr-paths.hpp.in.

25.31.1.8 #define LIBDIR "@libdir@"

Definition at line 10 of file sevmgr-paths.hpp.in.

25.31.1.9 #define LIBEXECDIR "@libexecdir@"

Definition at line 11 of file sevmgr-paths.hpp.in.

25.31.1.10 #define SBINDIR "@sbindir@"

Definition at line 12 of file sevmgr-paths.hpp.in.

25.31.1.11 #define SYSCONFDIR "@sysconfdir@"

Definition at line 13 of file sevmgr-paths.hpp.in.

25.31.1.12 #define INCLUDEDIR "@includedir@"

Definition at line 14 of file sevmgr-paths.hpp.in.

25.31.1.13 #define DATAROOTDIR "@datarootdir@"

Definition at line 15 of file sevmgr-paths.hpp.in.

25.31.1.14 #define DATADIR "@datadir@"

Definition at line 16 of file sevmgr-paths.hpp.in.

25.31.1.15 #define DOCDIR "@docdir@"

Definition at line 17 of file sevmgr-paths.hpp.in.

25.31.1.16 #define MANDIR "@mandir@"

Definition at line 18 of file sevmgr-paths.hpp.in.

25.31.1.17 #define INFODIR "@infodir@"

Definition at line 19 of file sevmgr-paths.hpp.in.

25.31.1.18 #define HTMLDIR "@htmldir@"

Definition at line 20 of file sevmgr-paths.hpp.in.

25.31.1.19 #define PDFDIR "@pdfdir@"

Definition at line 21 of file sevmgr-paths.hpp.in.

25.31.1.20 #define STDAIR_SAMPLE_DIR "@sampledir@"

Definition at line 22 of file sevmgr-paths.hpp.in.

25.32 sevmgr-paths.hpp.in

```
00001 #ifndef __SEVMGR_PATHS_HPP__
00002 #define __SEVMGR_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 // __SEVMGR_PATHS_HPP__
```

25.33 sevmgr/factory/FacSEVMGRServiceContext.cpp File Reference

```
#include <cassert> #include <stdair/service/FacSupervisor.-
hpp>   #include <sevmgr/factory/FacSEVMGRServiceContext.-
hpp>     #include <sevmgr/service/SEVMGR_ServiceContext.-
hpp>
```

Namespaces

- namespace SEVMGR

25.34 FacSEVMGRServiceContext.cpp

```
00001 // ////////////////////////////////////////////////////////////////////
00002 // Import section
00003 // ////////////////////////////////////////////////////////////////////
00004 // STL
```

```
00005 #include <cassert>
00006 // StdAir
00007 #include <stdair/service/FacSupervisor.hpp>
00008 // Sevmgr
00009 #include <sevmgr/factory/FacSEVMGRServiceContext.hpp>
00010 #include <sevmgr/service/SEVMGR_ServiceContext.hpp>
00011
00012 namespace SEVMGR {
00013
00014   FacSEVMGRServiceContext* FacSEVMGRServiceContext::_instance = NULL;
00015
00016   // ////////////////////////////////////////////////////////////////////
00017   FacSEVMGRServiceContext::~FacSEVMGRServiceContext () {
00018     _instance = NULL;
00019   }
00020
00021   // ////////////////////////////////////////////////////////////////////
00022   FacSEVMGRServiceContext& FacSEVMGRServiceContext::instance () {
00023
00024     if (_instance == NULL) {
00025       _instance = new FacSEVMGRServiceContext();
00026       assert (_instance != NULL);
00027
00028       stdair::FacSupervisor::instance().registerServiceFactory (_instance);
00029     }
00030     return *_instance;
00031   }
00032
00033   // ////////////////////////////////////////////////////////////////////
00034   SEVMGR_ServiceContext& FacSEVMGRServiceContext::create () {
00035     SEVMGR_ServiceContext* aServiceContext_ptr = NULL;
00036
00037     aServiceContext_ptr = new SEVMGR_ServiceContext ();
00038     assert (aServiceContext_ptr != NULL);
00039
00040     // The new object is added to the Bom pool
00041     _pool.push_back (aServiceContext_ptr);
00042
00043     return *aServiceContext_ptr;
00044   }
00045
00046 }
```

25.35 sevmgr/factory/FacSEVMGRServiceContext.hpp File Reference

`#include <stdair/service/FacServiceAbstract.hpp> #include <sevmgr/SEVMGR_Types.hpp>`

Classes

- class SEVMGR::FacSEVMGRServiceContext

Namespaces

- namespace SEVMGR

25.36 FacSEVMGRServiceContext.hpp

```
00001 #ifndef __SEVMGR_FAC_FACSEVMGRSERVICECONTEXT_HPP
00002 #define __SEVMGR_FAC_FACSEVMGRSERVICECONTEXT_HPP
00003
```

```
00004 // //////////////////////////////////////////////////////////////////
00005 // Import section
00006 // //////////////////////////////////////////////////////////////////
00007 // StdAir
00008 #include <stdair/service/FacServiceAbstract.hpp>
00009 // Sevmgr
00010 #include <sevmgr/SEVMGR_Types.hpp>
00011
00012 namespace SEVMGR {
00013
00015   class SEVMGR_ServiceContext;
00016
00018   class FacSEVMGRServiceContext : public stdair::FacServiceAbstract {
00019   public:
00020
00024     static FacSEVMGRServiceContext& instance();
00025
00030     ~FacSEVMGRServiceContext();
00031
00035     SEVMGR_ServiceContext& create ();
00036
00037
00038   protected:
00042     FacSEVMGRServiceContext () {}
00043
00044   private:
00046     static FacSEVMGRServiceContext* _instance;
00047   };
00048
00049 }
00050 #endif // __SEVMGR_FAC_FACSEVMGRSERVICECONTEXT_HPP
```

25.37 sevmgr/python/pysevmgr.cpp File Reference

`#include <cassert> #include <stdexcept> #include <fstream> #include <sstream> #include <string> #include <list> #include <vector> #include <boost/python.hpp> #include <stdair/stdair_basic_types.hpp> #include <stdair/stdair_exceptions.hpp> #include <stdair/basic/BasFileMgr.hpp> #include <stdair/basic/BasLogParams.hpp> #include <stdair/basic/BasDBParams.hpp> #include <sevmgr/SEVMGR_Service.hpp>`

Classes

- struct SEVMGR::PYEventQueueManager

Namespaces

- namespace SEVMGR

Functions

- BOOST_PYTHON_MODULE (libpysevmgr)

25.37.1 Function Documentation

25.37.1.1 BOOST_PYTHON_MODULE (libpysevmgr)

Definition at line 152 of file pysevmgr.cpp.

References SEVMGR::PYEventQueueManager::sevmgr(), and SEVMGR::PYEventQueueManager::init().

25.38 pysevmgr.cpp

```
00001 // STL
00002 #include <cassert>
00003 #include <stdexcept>
00004 #include <fstream>
00005 #include <sstream>
00006 #include <string>
00007 #include <list>
00008 #include <vector>
00009 // Boost String
00010 #include <boost/python.hpp>
00011 // StdAir
00012 #include <stdair/stdair_basic_types.hpp>
00013 #include <stdair/stdair_exceptions.hpp>
00014 #include <stdair/basic/BasFileMgr.hpp>
00015 #include <stdair/basic/BasLogParams.hpp>
00016 #include <stdair/basic/BasDBParams.hpp>
00017 // SEvMgr
00018 #include <sevmgr/SEVMGR_Service.hpp>
00019
00020 namespace SEVMGR {
00021
00022   struct PYEventQueueManager {
00023   public:
00025     std::string sevmgr() {
00026       std::ostringstream oStream;
00027
00028       // Sanity check
00029       if (_logOutputStream == NULL) {
00030         oStream << "The log filepath is not valid." << std::endl;
00031         return oStream.str();
00032       }
00033       assert (_logOutputStream != NULL);
00034
00035       try {
00036
00037         // DEBUG
00038         *_logOutputStream << "Default service" << std::endl;
00039
00040         if (_sevmgrService == NULL) {
00041           oStream << "The Sevmgr service has not been initialised, "
00042                   << "i.e., the init() method has not been called "
00043                   << "correctly on the PYEventQueueManager object. Please "
00044                   << "check that all the parameters are not empty and "
00045                   << "point to actual files.";
00046           *_logOutputStream << oStream.str();
00047           return oStream.str();
00048         }
00049         assert (_sevmgrService != NULL);
00050
00051         // Do the sevmgr
00052         _sevmgrService->buildSampleBom();
00053
00054         // DEBUG
00055         *_logOutputStream << "Default service returned" << std::endl;
00056
00057         // DEBUG
00058         *_logOutputStream << "Sevmgr output: " << oStream.str() << std::endl;
00059
00060       } catch (const stdair::RootException& eSevmgrError) {
00061         *_logOutputStream << "Sevmgr error: "  << eSevmgrError.what()
00062                           << std::endl;
```

```
00063
00064       } catch (const std::exception& eStdError) {
00065         *_logOutputStream << "Error: "  << eStdError.what() << std::endl;
00066
00067       } catch (...) {
00068         *_logOutputStream << "Unknown error" << std::endl;
00069       }
00070
00071       return oStream.str();
00072     }
00073
00074   public:
00076     PYEventQueueManager() : _sevmgrService (NULL), _logOutputStream (NULL) {
00077     }
00078
00080     PYEventQueueManager (const PYEventQueueManager& iPYEventQueueManager)
00081       : _sevmgrService (iPYEventQueueManager._sevmgrService),
00082         _logOutputStream (iPYEventQueueManager._logOutputStream) {
00083     }
00084
00086     ~PYEventQueueManager() {
00087       _sevmgrService = NULL;
00088       _logOutputStream = NULL;
00089     }
00090
00092     bool init (const std::string& iLogFilepath,
00093                const std::string& iDBUser, const std::string& iDBPasswd,
00094                const std::string& iDBHost, const std::string& iDBPort,
00095                const std::string& iDBDBName) {
00096       bool isEverythingOK = true;
00097
00098       try {
00099
00100         // Check that the file path given as input corresponds to an actual
      file
00101         const bool isWriteable = (iLogFilepath.empty() == false);
00102         // stdair::BasFileMgr::isWriteable (iLogFilepath);
00103         if (isWriteable == false) {
00104           isEverythingOK = false;
00105           return isEverythingOK;
00106         }
00107
00108         // Set the log parameters
00109         _logOutputStream = new std::ofstream;
00110         assert (_logOutputStream != NULL);
00111
00112         // Open and clean the log outputfile
00113         _logOutputStream->open (iLogFilepath.c_str());
00114         _logOutputStream->clear();
00115
00116         // DEBUG
00117         *_logOutputStream << "Python wrapper initialisation" << std::endl;
00118         const stdair::BasLogParams lLogParams (stdair::LOG::DEBUG,
00119                                                *_logOutputStream);
00120
00121         // Initialise the context
00122         stdair::BasDBParams lDBParams (iDBUser, iDBPasswd, iDBHost, iDBPort,
00123                                        iDBDBName);
00124         _sevmgrService = new SEVMGR_Service (lLogParams, lDBParams);
00125
00126         // DEBUG
00127         *_logOutputStream << "Python wrapper initialised" << std::endl;
00128
00129       } catch (const stdair::RootException& eSevmgrError) {
00130         *_logOutputStream << "Sevmgr error: "  << eSevmgrError.what()
00131                           << std::endl;
00132
00133       } catch (const std::exception& eStdError) {
00134         *_logOutputStream << "Error: "  << eStdError.what() << std::endl;
00135
00136       } catch (...) {
00137         *_logOutputStream << "Unknown error" << std::endl;
00138       }
00139
```

```
00140       return isEverythingOK;
00141     }
00142
00143   private:
00145     SEVMGR_Service* _sevmgrService;
00146     std::ofstream* _logOutputStream;
00147   };
00148
00149 }
00150
00151 // ////////////////////////////////////////////////////////////
00152 BOOST_PYTHON_MODULE(libpysevmgr) {
00153   boost::python::class_<SEVMGR::PYEventQueueManager> ("PYEventQueueManager")
00154     .def ("sevmgr", &SEVMGR::PYEventQueueManager::sevmgr)
00155     .def ("init", &SEVMGR::PYEventQueueManager::init);
00156 }
```

25.39 sevmgr/service/SEVMGR_Service.cpp File Reference

`#include <cassert> #include <sstream> #include <boost/make_shared.hpp> #include <stdair/basic/BasChronometer.hpp> #include <stdair/basic/BasConst_General.hpp> #include <stdair/bom/BomRoot.hpp> #include <stdair/bom/BomDisplay.hpp> #include <stdair/bom/EventStruct.hpp> #include <stdair/bom/EventQueue.hpp> #include <stdair/service/Logger.hpp> #include <stdair/STDAIR_Service.hpp> #include <sevmgr/basic/BasConst_SEVMGR_Service.hpp> #include <sevmgr/factory/FacSEVMGRServiceContext.hpp> #include <sevmgr/command/EventQueueManager.hpp> #include <sevmgr/service/SEVMGR_ServiceContext.hpp> #include <sevmgr/SEVMGR_Service.hpp>`

Namespaces

- namespace SEVMGR

25.40 SEVMGR_Service.cpp

```
00001 // ////////////////////////////////////////////////////////////////////
00002 // Import section
00003 // ////////////////////////////////////////////////////////////////////
00004 // STL
00005 #include <cassert>
00006 #include <sstream>
00007 // Boost
00008 #include <boost/make_shared.hpp>
00009 // StdAir
00010 #include <stdair/basic/BasChronometer.hpp>
00011 #include <stdair/basic/BasConst_General.hpp>
00012 #include <stdair/bom/BomRoot.hpp>
00013 #include <stdair/bom/BomDisplay.hpp>
00014 #include <stdair/bom/EventStruct.hpp>
00015 #include <stdair/bom/EventQueue.hpp>
00016 #include <stdair/service/Logger.hpp>
00017 #include <stdair/STDAIR_Service.hpp>
00018 // Sevmgr
00019 #include <sevmgr/basic/BasConst_SEVMGR_Service.hpp>
00020 #include <sevmgr/factory/FacSEVMGRServiceContext.hpp>
00021 #include <sevmgr/command/EventQueueManager.hpp>
```

```
00022 #include <sevmgr/service/SEVMGR_ServiceContext.hpp>
00023 #include <sevmgr/SEVMGR_Service.hpp>
00024
00025 namespace SEVMGR {
00026
00027   // ////////////////////////////////////////////////////////////////////
00028   SEVMGR_Service::SEVMGR_Service() : _sevmgrServiceContext (NULL) {
00029     assert (false);
00030   }
00031
00032   // ////////////////////////////////////////////////////////////////////
00033   SEVMGR_Service::SEVMGR_Service (const SEVMGR_Service& iService)
00034     : _sevmgrServiceContext (NULL) {
00035     assert (false);
00036   }
00037
00038   // ////////////////////////////////////////////////////////////////////
00039   SEVMGR_Service::SEVMGR_Service (const stdair::BasLogParams& iLogParams,
00040                                   const stdair::BasDBParams& iDBParams)
00041     : _sevmgrServiceContext (NULL) {
00042
00043     // Initialise the STDAIR service handler
00044     stdair::STDAIR_ServicePtr_T lSTDAIR_Service_ptr =
00045       initStdAirService (iLogParams, iDBParams);
00046
00047     // Initialise the service context
00048     initServiceContext();
00049
00050     // Add the StdAir service context to the SEVMGR service context
00051     // \note SEVMGR owns the STDAIR service resources here.
00052     const bool ownStdairService = true;
00053     addStdAirService (lSTDAIR_Service_ptr, ownStdairService);
00054
00055     // Initialise the (remaining of the) context
00056     initSevmgrService();
00057   }
00058
00059   // ////////////////////////////////////////////////////////////////////
00060   SEVMGR_Service::SEVMGR_Service (const stdair::BasLogParams& iLogParams)
00061     : _sevmgrServiceContext (NULL) {
00062
00063     // Initialise the STDAIR service handler
00064     stdair::STDAIR_ServicePtr_T lSTDAIR_Service_ptr =
00065       initStdAirService (iLogParams);
00066
00067     // Initialise the service context
00068     initServiceContext();
00069
00070     // Add the StdAir service context to the SEVMGR service context
00071     // \note SEVMGR owns the STDAIR service resources here.
00072     const bool ownStdairService = true;
00073     addStdAirService (lSTDAIR_Service_ptr, ownStdairService);
00074
00075     // Initialise the (remaining of the) context
00076     initSevmgrService();
00077   }
00078
00079   // ////////////////////////////////////////////////////////////////////
00080   SEVMGR_Service::
00081   SEVMGR_Service (stdair::STDAIR_ServicePtr_T ioSTDAIR_Service_ptr)
00082     : _sevmgrServiceContext (NULL) {
00083
00084     // Initialise the service context
00085     initServiceContext();
00086
00087     // Add the StdAir service context to the SEVMGR service context
00088     // \note Sevmgr does not own the STDAIR service resources here.
00089     const bool doesNotOwnStdairService = false;
00090     addStdAirService (ioSTDAIR_Service_ptr, doesNotOwnStdairService);
00091
00092     // Initialise the context
00093     initSevmgrService();
00094   }
00095
```

```
00096   // ////////////////////////////////////////////////////////////////////
00097   SEVMGR_Service::~SEVMGR_Service() {
00098     // Delete/Clean all the objects from memory
00099     finalise();
00100   }
00101
00102   // ////////////////////////////////////////////////////////////////////
00103   void SEVMGR_Service::finalise() {
00104     assert (_sevmgrServiceContext != NULL);
00105     // Reset the (Boost.)Smart pointer pointing on the STDAIR_Service object.
00106     _sevmgrServiceContext->reset();
00107   }
00108
00109   // ////////////////////////////////////////////////////////////////////
00110   void SEVMGR_Service::initServiceContext() {
00111     // Initialise the service context
00112     SEVMGR_ServiceContext& lSEVMGR_ServiceContext =
00113       FacSEVMGRServiceContext::instance().create();
00114     _sevmgrServiceContext = &lSEVMGR_ServiceContext;
00115   }
00116
00117   // ////////////////////////////////////////////////////////////////////
00118   void SEVMGR_Service::
00119   addStdAirService (stdair::STDAIR_ServicePtr_T ioSTDAIR_Service_ptr,
00120                     const bool iOwnStdairService) {
00121     // Retrieve the Sevmgr service context
00122     assert (_sevmgrServiceContext != NULL);
00123     SEVMGR_ServiceContext& lSEVMGR_ServiceContext =
00124       *_sevmgrServiceContext;
00125
00126     // Store the STDAIR service object within the (SEVMGR) service context
00127     lSEVMGR_ServiceContext.setSTDAIR_Service (ioSTDAIR_Service_ptr,
00128                                               iOwnStdairService);
00129   }
00130
00131   // ////////////////////////////////////////////////////////////////////
00132   stdair::STDAIR_ServicePtr_T SEVMGR_Service::
00133   initStdAirService (const stdair::BasLogParams& iLogParams,
00134                      const stdair::BasDBParams& iDBParams) {
00135
00141     stdair::STDAIR_ServicePtr_T lSTDAIR_Service_ptr =
00142       boost::make_shared<stdair::STDAIR_Service> (iLogParams, iDBParams);
00143     assert (lSTDAIR_Service_ptr != NULL);
00144
00145     return lSTDAIR_Service_ptr;
00146   }
00147
00148   // ////////////////////////////////////////////////////////////////////
00149   stdair::STDAIR_ServicePtr_T SEVMGR_Service::
00150   initStdAirService (const stdair::BasLogParams& iLogParams) {
00151
00157     stdair::STDAIR_ServicePtr_T lSTDAIR_Service_ptr =
00158       boost::make_shared<stdair::STDAIR_Service> (iLogParams);
00159     assert (lSTDAIR_Service_ptr != NULL);
00160
00161     return lSTDAIR_Service_ptr;
00162   }
00163
00164   // ////////////////////////////////////////////////////////////////////
00165   void SEVMGR_Service::initSevmgrService() {
00166     // Do nothing at this stage. A sample BOM tree may be built by
00167     // calling the buildSampleBom() method
00168   }
00169
00170   // ////////////////////////////////////////////////////////////////////
00171   void SEVMGR_Service::buildSampleBom() {
00172
00173     // Retrieve the Sevmgr service context
00174     if (_sevmgrServiceContext == NULL) {
00175       throw stdair::NonInitialisedServiceException ("The Sevmgr service has "
00176                                                     "not been initialised");
00177     }
00178     assert (_sevmgrServiceContext != NULL);
00179
```

```
00180     //SEVMGR_ServiceContext& lSEVMGR_ServiceContext = *_sevmgrServiceContext;
00181
00182     // Retrieve the STDAIR service object from the (Sevmgr) service context
00183     //stdair::STDAIR_Service& lSTDAIR_Service =
00184     // lSEVMGR_ServiceContext.getSTDAIR_Service();
00185
00186     // Retrieve the event queue
00187     //stdair::EventQueue& lEventQueue = lSTDAIR_Service.getEventQueue();
00188   }
00189
00190   // ////////////////////////////////////////////////////////////////////
00191   std::string SEVMGR_Service::csvDisplay() const {
00192
00193     // Retrieve the Sevmgr service context
00194     if (_sevmgrServiceContext == NULL) {
00195       throw stdair::NonInitialisedServiceException ("The Sevmgr service has "
00196                                                     "not been initialised");
00197     }
00198     assert (_sevmgrServiceContext != NULL);
00199
00200     SEVMGR_ServiceContext& lSEVMGR_ServiceContext = *_sevmgrServiceContext;
00201
00202     // Retrieve the STDAIR service object from the (Sevmgr) service context
00203     stdair::STDAIR_Service& lSTDAIR_Service =
00204       lSEVMGR_ServiceContext.getSTDAIR_Service();
00205
00206     // Retrieve the event queue
00207     stdair::EventQueue& lEventQueue = lSTDAIR_Service.getEventQueue();
00208
00209     // Delegate the BOM building to the dedicated service
00210     return stdair::BomDisplay::csvDisplay (lEventQueue);
00211   }
00212
00213   // ////////////////////////////////////////////////////////////////////
00214   stdair::ProgressStatusSet SEVMGR_Service::
00215   popEvent (stdair::EventStruct& ioEventStruct) const {
00216
00217     // Retrieve the Sevmgr service context
00218     assert (_sevmgrServiceContext != NULL);
00219     SEVMGR_ServiceContext& lSEVMGR_ServiceContext = *_sevmgrServiceContext;
00220
00221     // Retrieve the StdAir service context
00222     stdair::STDAIR_Service& lSTDAIR_Service =
00223       lSEVMGR_ServiceContext.getSTDAIR_Service();
00224
00225     // Retrieve the event queue object instance
00226     stdair::EventQueue& lQueue = lSTDAIR_Service.getEventQueue();
00227
00228     // Extract the next event from the queue
00229     return lQueue.popEvent (ioEventStruct);
00230   }
00231
00232   // ////////////////////////////////////////////////////////////////////
00233   bool SEVMGR_Service::isQueueDone() const {
00234
00235     // Retrieve the Sevmgr service context
00236     assert (_sevmgrServiceContext != NULL);
00237     SEVMGR_ServiceContext& lSEVMGR_ServiceContext =
00238       *_sevmgrServiceContext;
00239
00240     // Retrieve the StdAir service context
00241     stdair::STDAIR_Service& lSTDAIR_Service =
00242       lSEVMGR_ServiceContext.getSTDAIR_Service();
00243
00244     // Retrieve the event queue object instance
00245     const stdair::EventQueue& lQueue = lSTDAIR_Service.getEventQueue();
00246
00247     // Calculates whether the event queue has been fully emptied
00248     const bool isQueueDone = lQueue.isQueueDone();
00249
00250     //
00251     return isQueueDone;
00252   }
00253
```

```
00254   // ////////////////////////////////////////////////////////////////////
00255   void SEVMGR_Service::reset() const {
00256
00257     // Retrieve the Sevmgr service context
00258     assert (_sevmgrServiceContext != NULL);
00259     SEVMGR_ServiceContext& lSEVMGR_ServiceContext =
00260       *_sevmgrServiceContext;
00261
00262     // Retrieve the StdAir service context
00263     stdair::STDAIR_Service& lSTDAIR_Service =
00264       lSEVMGR_ServiceContext.getSTDAIR_Service();
00265     // Retrieve the event queue object instance
00266     stdair::EventQueue& lQueue = lSTDAIR_Service.getEventQueue();
00267
00268     // Delegate the call to the dedicated command
00269     EventQueueManager::reset (lQueue);
00270   }
00271 }
```

25.41 sevmgr/service/SEVMGR_ServiceContext.cpp File Reference

`#include <cassert> #include <sstream> #include <stdair/STDAIR_Service.hpp> #include <stdair/basic/BasConst_General.hpp> #include <sevmgr/basic/BasConst_EventQueueManager.hpp> #include <sevmgr/service/SEVMGR_ServiceContext.hpp>`

Namespaces

- namespace SEVMGR

25.42 SEVMGR_ServiceContext.cpp

```
00001 // ////////////////////////////////////////////////////////////////////
00002 // Import section
00003 // ////////////////////////////////////////////////////////////////////
00004 // STL
00005 #include <cassert>
00006 #include <sstream>
00007 // StdAir
00008 #include <stdair/STDAIR_Service.hpp>
00009 #include <stdair/basic/BasConst_General.hpp>
00010 // SEvMgr
00011 #include <sevmgr/basic/BasConst_EventQueueManager.hpp>
00012 #include <sevmgr/service/SEVMGR_ServiceContext.hpp>
00013
00014 namespace SEVMGR {
00015
00016   // ////////////////////////////////////////////////////////////////////
00017   SEVMGR_ServiceContext::SEVMGR_ServiceContext() {
00018   }
00019
00020   // ////////////////////////////////////////////////////////////////////
00021   SEVMGR_ServiceContext::
00022   SEVMGR_ServiceContext (const std::string& iServiceName) {
00023   }
00024
00025   // ////////////////////////////////////////////////////////////////////
00026   SEVMGR_ServiceContext::~SEVMGR_ServiceContext() {
00027   }
00028
00029   // ////////////////////////////////////////////////////////////////////
00030   const std::string SEVMGR_ServiceContext::shortDisplay() const {
```

```
00031     std::ostringstream oStr;
00032     oStr << "SEVMGR_ServiceContext -- Owns StdAir service: "
00033          << _ownStdairService;
00034     return oStr.str();
00035   }
00036
00037   // ////////////////////////////////////////////////////////////////////
00038   const std::string SEVMGR_ServiceContext::display() const {
00039     std::ostringstream oStr;
00040     oStr << shortDisplay();
00041     return oStr.str();
00042   }
00043
00044   // ////////////////////////////////////////////////////////////////////
00045   const std::string SEVMGR_ServiceContext::describe() const {
00046     return shortDisplay();
00047   }
00048
00049   // ////////////////////////////////////////////////////////////////////
00050   void SEVMGR_ServiceContext::reset() {
00051     if (_ownStdairService == true) {
00052       _stdairService.reset();
00053     }
00054   }
00055
00056 }
```

25.43 sevmgr/service/SEVMGR_ServiceContext.hpp File Reference

`#include <string> #include <stdair/stdair_basic_types.hpp> #include <stdair/stdair_service_types.hpp> #include <stdair/service/ServiceAbstract.hpp> #include <sevmgr/SEVMGR_Types.hpp>`

Classes

- class SEVMGR::SEVMGR_ServiceContext

  Class holding the context of the Sevmgr services.

Namespaces

- namespace SEVMGR

25.44 SEVMGR_ServiceContext.hpp

```
00001 #ifndef __SEVMGR_SVC_SEVMGRSERVICECONTEXT_HPP
00002 #define __SEVMGR_SVC_SEVMGRSERVICECONTEXT_HPP
00003
00004 // ////////////////////////////////////////////////////////////////////
00005 // Import section
00006 // ////////////////////////////////////////////////////////////////////
00007 // STL
00008 #include <string>
00009 // StdAir
00010 #include <stdair/stdair_basic_types.hpp>
00011 #include <stdair/stdair_service_types.hpp>
00012 #include <stdair/service/ServiceAbstract.hpp>
00013 // SEvMgr
00014 #include <sevmgr/SEVMGR_Types.hpp>
```

```
00015
00016 namespace SEVMGR {
00017
00021   class SEVMGR_ServiceContext : public stdair::ServiceAbstract {
00027     friend class SEVMGR_Service;
00028     friend class FacSEVMGRServiceContext;
00029
00030   private:
00031     // ///////// Getters //////////
00035     stdair::STDAIR_ServicePtr_T getSTDAIR_ServicePtr() const {
00036       return _stdairService;
00037     }
00038
00042     stdair::STDAIR_Service& getSTDAIR_Service() const {
00043       assert (_stdairService != NULL);
00044       return *_stdairService;
00045     }
00046
00050     const bool getOwnStdairServiceFlag() const {
00051       return _ownStdairService;
00052     }
00053
00054
00055   private:
00056     // ///////// Setters //////////
00060     void setSTDAIR_Service (stdair::STDAIR_ServicePtr_T ioSTDAIR_ServicePtr,
00061                             const bool iOwnStdairService) {
00062       _stdairService = ioSTDAIR_ServicePtr;
00063       _ownStdairService = iOwnStdairService;
00064     }
00065
00066
00067   private:
00068     // ///////// Display Methods //////////
00072     const std::string shortDisplay() const;
00073
00077     const std::string display() const;
00078
00082     const std::string describe() const;
00083
00084
00085   private:
00087
00090     SEVMGR_ServiceContext();
00091     SEVMGR_ServiceContext (const std::string& iServiceName);
00095     SEVMGR_ServiceContext (const SEVMGR_ServiceContext&);
00096
00100     ~SEVMGR_ServiceContext();
00101
00105     void reset();
00106
00107
00108   private:
00109     // /////////////// Children ///////////////
00113     stdair::STDAIR_ServicePtr_T _stdairService;
00114
00118     bool _ownStdairService;
00119
00120
00121   private:
00122     // ////////////// Attributes ////////////////
00123     // No attributes for now
00124   };
00125
00126 }
00127 #endif // __SEVMGR_SVC_SEVMGRSERVICECONTEXT_HPP
```

25.45 sevmgr/SEVMGR_Exceptions.hpp File Reference

`#include <exception> #include <stdair/stdair_exceptions.-`

hpp>

Classes

- class SEVMGR::SEvMgrException

Namespaces

- namespace SEVMGR

25.46 SEVMGR_Exceptions.hpp

```
00001 #ifndef __SEVMGR_SEVMGR_EXCEPTIONS_HPP
00002 #define __SEVMGR_SEVMGR_EXCEPTIONS_HPP
00003
00004 // //////////////////////////////////////////////////////////////////
00005 // Import section
00006 // //////////////////////////////////////////////////////////////////
00007 // STL
00008 #include <exception>
00009 // StdAir
00010 #include <stdair/stdair_exceptions.hpp>
00011
00012 namespace SEVMGR {
00013
00014   // ///////// Exceptions ////////////
00018   class SEvMgrException : public stdair::RootException {
00019   public:
00023     SEvMgrException (const std::string& iWhat)
00024       : stdair::RootException (iWhat) {}
00025   };
00026
00027 }
00028 #endif // __SEVMGR_SEVMGR_EXCEPTIONS_HPP
00029
```

25.47 sevmgr/SEVMGR_Service.hpp File Reference

#include <stdair/stdair_basic_types.hpp> #include <stdair/stdair-_service_types.hpp> #include <stdair/bom/EventTypes.-hpp>

Classes

- class SEVMGR::SEVMGR_Service

  class holding the services related to Travel Demand Generation.

Namespaces

- namespace stdair
- namespace SEVMGR

25.48 SEVMGR_Service.hpp

```
00001 #ifndef __SEVMGR_SEVMGR_SERVICE_HPP
00002 #define __SEVMGR_SEVMGR_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/bom/EventTypes.hpp>
00011
00012 // Forward declarations
00013 namespace stdair {
00014   class EventQueue;
00015   struct ProgressStatusSet;
00016   struct BasLogParams;
00017   struct BasDBParams;
00018   struct EventStruct;
00019 }
00020
00021 namespace SEVMGR {
00022
00024   class SEVMGR_ServiceContext;
00025
00029   class SEVMGR_Service {
00030   public:
00031     // ////////////////// Constructors and Destructors /////////////////
00047     SEVMGR_Service (const stdair::BasLogParams&, const stdair::BasDBParams&);
00048
00060     SEVMGR_Service (const stdair::BasLogParams&);
00061
00077     SEVMGR_Service (stdair::STDAIR_ServicePtr_T);
00078
00082     ~SEVMGR_Service();
00083
00084
00085   public:
00086     // ////////////////// Business support methods /////////////////
00090     void buildSampleBom();
00091
00108     stdair::ProgressStatusSet popEvent (stdair::EventStruct&) const;
00109
00115     bool isQueueDone() const;
00116
00121     void reset() const;
00122
00123
00124   public:
00125     // ////////////////// Display support methods /////////////////
00133     std::string csvDisplay() const;
00134
00135
00136   private:
00137     // ////////////////// Constructors and Destructors /////////////////
00141     SEVMGR_Service();
00142
00146     SEVMGR_Service (const SEVMGR_Service&);
00147
00152     void initServiceContext();
00153
00165     stdair::STDAIR_ServicePtr_T initStdAirService (const stdair::BasLogParams&,
00166                                                    const stdair::BasDBParams&);
00167
00177     stdair::STDAIR_ServicePtr_T initStdAirService (const stdair::BasLogParams&)
      ;
00178
00187     void addStdAirService (stdair::STDAIR_ServicePtr_T ioSTDAIR_ServicePtr,
00188                            const bool iOwnStdairService);
00189
00196     void initSevmgrService();
00197
```

```
00201     void finalise();
00202
00203
00204   private:
00205     // ///////// Service Context /////////
00209     SEVMGR_ServiceContext* _sevmgrServiceContext;
00210   };
00211
00212 }
00213 #endif // __SEVMGR_SEVMGR_SERVICE_HPP
```

25.49 sevmgr/SEVMGR_Types.hpp File Reference

`#include <boost/shared_ptr.hpp> #include <sevmgr/SEVMGR_-Exceptions.hpp>`

Namespaces

- namespace SEVMGR

Typedefs

- typedef boost::shared_ptr < SEVMGR_Service > SEVMGR::SEVMGR_-ServicePtr_T

25.50 SEVMGR_Types.hpp

```
00001 #ifndef __SEVMGR_SEVMGR_TYPES_HPP
00002 #define __SEVMGR_SEVMGR_TYPES_HPP
00003
00004 // ////////////////////////////////////////////////////////////////////
00005 // Import section
00006 // ////////////////////////////////////////////////////////////////////
00007 // Boost
00008 #include <boost/shared_ptr.hpp>
00009 // Sevmgr
00010 #include <sevmgr/SEVMGR_Exceptions.hpp>
00011
00012 namespace SEVMGR {
00013
00014   // Forward declarations
00015   class SEVMGR_Service;
00016
00017   // //////// Type definitions specific to to Sevmgr /////////
00021   typedef boost::shared_ptr<SEVMGR_Service> SEVMGR_ServicePtr_T;
00022
00023 }
00024 #endif // __SEVMGR_SEVMGR_TYPES_HPP
00025
```

25.51 sevmgr/ui/cmdline/readline_autocomp.hpp File Reference

`#include <string> #include <iosfwd> #include <cstdio> × #include <sys/types.h> #include <sys/file.h> #include <sys/stat.h> #include <sys/errno.h> #include <readline/readline.-h> #include <readline/history.h>`

Classes

- struct COMMAND

Typedefs

- typedef int(* pt2Func)(char *)

Functions

- char * getwd ()
- char * xmalloc (size_t)
- int com_list (char *)
- int com_view (char *)
- int com_rename (char *)
- int com_stat (char *)
- int com_pwd (char *)
- int com_delete (char *)
- int com_help (char *)
- int com_cd (char *)
- int com_quit (char *)
- char * stripwhite (char *iString)
- COMMAND * find_command (char *iString)
- char * dupstr (char *iString)
- int execute_line (char *line)
- char * command_generator (char *text, int state)
- char ** fileman_completion (char *text, int start, int end)
- void initialize_readline ()
- void too_dangerous (char *caller)
- int valid_argument (char *caller, char *arg)

Variables

- COMMAND commands []
- int done
- static char syscom [1024]

25.51.1 Typedef Documentation

25.51.1.1 typedef int(* pt2Func)(char *)

Definition at line 35 of file readline_autocomp.hpp.

25.51.2 Function Documentation

25.51.2.1 char* getwd ( )

readline_autocomp.hpp -- A tiny application which demonstrates how to use the GNU Readline library. This application interactively allows users to manipulate files and their modes.

Referenced by com_pwd().

25.51.2.2 char* xmalloc (size_t)

Referenced by dupstr().

25.51.2.3 void com_list (char * *arg*)

List the file(s) named in arg.

Definition at line 264 of file readline_autocomp.hpp.

25.51.2.4 int com_view (char * *arg*)

Definition at line 274 of file readline_autocomp.hpp.

References valid_argument().

25.51.2.5 int com_rename (char * *arg*)

Definition at line 284 of file readline_autocomp.hpp.

References too_dangerous().

25.51.2.6 int com_stat (char * *arg*)

Definition at line 289 of file readline_autocomp.hpp.

References valid_argument().

25.51.2.7 int com_pwd (char * *ignore*)

Definition at line 367 of file readline_autocomp.hpp.

References getwd().

Referenced by com_cd().

25.51.2.8 int com_delete (char * *arg*)

Definition at line 315 of file readline_autocomp.hpp.

References too_dangerous().

25.51.2.9 int com_help (char * *arg*)

Print out help for ARG, or for all of the commands if ARG is not present.

Definition at line 324 of file readline_autocomp.hpp.

References COMMAND::name.

25.51.2.10 int com_cd (char * *arg*)

Definition at line 356 of file readline_autocomp.hpp.

References com_pwd().

25.51.2.11 int com_quit (char * *arg*)

Definition at line 381 of file readline_autocomp.hpp.

25.51.2.12 char * stripwhite (char * *string*)

Strip whitespace from the start and end of STRING. Return a pointer into STRING.

Definition at line 152 of file readline_autocomp.hpp.

25.51.2.13 COMMAND * find_command (char * *name*)

Look up NAME as the name of a command, and return a pointer to that command. Return a NULL pointer if NAME isn't a command name.

Definition at line 136 of file readline_autocomp.hpp.

References COMMAND::name.

Referenced by execute_line().

25.51.2.14 char* dupstr (char * *iString*)

Duplicate a string

Definition at line 85 of file readline_autocomp.hpp.

References xmalloc().

Referenced by command_generator().

25.51.2.15 int execute_line (char * *line*)

Execute a command line.

Definition at line 94 of file readline_autocomp.hpp.

References find_command(), and COMMAND::func.

25.51.2.16 char * command_generator (char * *text,* int *state*)

Generator function for command completion. STATE lets us know whether to start from scratch; without any state (i.e. STATE == 0), then we start at the top of the list.

Definition at line 222 of file readline_autocomp.hpp.

References dupstr().

Referenced by fileman_completion().

25.51.2.17 char ** fileman_completion (char * *text*, int *start*, int *end*)

Attempt to complete on the contents of TEXT. START and END bound the region of rl_line_buffer that contains the word to complete. TEXT is the word to complete. We can use the entire contents of rl_line_buffer in case we want to do some simple parsing. Return the array of matches, or NULL if there aren't any.

Definition at line 200 of file readline_autocomp.hpp.

References command_generator().

Referenced by initialize_readline().

25.51.2.18 void initialize_readline ( )

Tell the GNU Readline library how to complete. We want to try to complete on command names if this is the first word in the line, or on filenames if not.

Definition at line 185 of file readline_autocomp.hpp.

References fileman_completion().

25.51.2.19 void too_dangerous (char * *caller*)

Definition at line 387 of file readline_autocomp.hpp.

Referenced by com_rename(), and com_delete().

25.51.2.20 int valid_argument (char * *caller*, char * *arg*)

Definition at line 395 of file readline_autocomp.hpp.

Referenced by com_view(), and com_stat().

25.51.3 Variable Documentation

25.51.3.1 COMMAND commands[]

Initial value:

```
{
  { "cd", (*com_cd)(), "Change to directory DIR" },
  { "delete", com_delete, "Delete FILE" },
  { "help", com_help, "Display this text" },
  { "?", com_help, "Synonym for `help'" },
  { "list", com_list, "List files in DIR" },
  { "ls", com_list, "Synonym for `list'" },
  { "pwd", com_pwd, "Print the current working directory" },
  { "quit", com_quit, "Quit using airinv" },
  { "rename", com_rename, "Rename FILE to NEWNAME" },
  { "stat", com_stat, "Print out statistics on FILE" },
  { "view", com_view, "View the contents of FILE" },
  { (char*) NULL, (pt2Func) NULL, (char*) NULL }
}
```

Definition at line 58 of file readline_autocomp.hpp.

25.51.3.2 int done

When non-zero, this global means the user is done using this program.

Definition at line 80 of file readline_autocomp.hpp.

25.51.3.3 char syscom[1024] [static]

String to pass to system(). This is for the LIST, VIEW and RENAME commands.

Definition at line 259 of file readline_autocomp.hpp.

25.52 readline_autocomp.hpp

```
00001
00006 #ifndef __AIRINV_READLINE_AUTOCOMP_HPP
00007 #define __AIRINV_READLINE_AUTOCOMP_HPP
00008
00009 // STL
00010 #include <string>
00011 #include <iosfwd>
00012 #include <cstdio>
00013 #include <sys/types.h>
00014 #include <sys/file.h>
00015 #include <sys/stat.h>
00016 #include <sys/errno.h>
00017
00018 #include <readline/readline.h>
00019 #include <readline/history.h>
00020
00021 extern char* getwd();
00022 extern char* xmalloc (size_t);
00023
00024 /* The names of functions that actually do the manipulation. */
00025 int com_list (char*);
00026 int com_view (char*);
00027 int com_rename (char*);
00028 int com_stat (char*);
00029 int com_pwd (char*);
00030 int com_delete (char*);
00031 int com_help (char*);
00032 int com_cd (char*);
00033 int com_quit (char*);
00034
00035 typedef int (*pt2Func) (char*);
00036
00041 typedef struct {
00045   char const* name;
00046
00050   pt2Func *func;
00051
00055   char *doc;
00056 } COMMAND;
00057
00058 COMMAND commands[] = {
00059   { "cd", (*com_cd)(), "Change to directory DIR" },
00060   { "delete", com_delete, "Delete FILE" },
00061   { "help", com_help, "Display this text" },
00062   { "?", com_help, "Synonym for `help'" },
00063   { "list", com_list, "List files in DIR" },
00064   { "ls", com_list, "Synonym for `list'" },
00065   { "pwd", com_pwd, "Print the current working directory" },
00066   { "quit", com_quit, "Quit using airinv" },
00067   { "rename", com_rename, "Rename FILE to NEWNAME" },
00068   { "stat", com_stat, "Print out statistics on FILE" },
00069   { "view", com_view, "View the contents of FILE" },
00070   { (char*) NULL, (pt2Func) NULL, (char*) NULL }
00071 };
```

```
00072
00073 // Forward declarations
00074 char* stripwhite (char* iString);
00075 COMMAND* find_command (char* iString);
00076
00080 int done;
00081
00085 char* dupstr (char* iString) {
00086   char* r = xmalloc (std::strlen (iString) + 1);
00087   strcpy (r, iString);
00088   return r;
00089 }
00090
00094 int execute_line (char* line) {
00095   register int i;
00096   COMMAND* command;
00097   char* word;
00098
00099   /* Isolate the command word. */
00100   i = 0;
00101   while (line[i] && whitespace (line[i])) {
00102     i++;
00103   }
00104   word = line + i;
00105
00106   while (line[i] && !whitespace (line[i])) {
00107     i++;
00108   }
00109
00110   if (line[i]) {
00111     line[i++] = '\0';
00112   }
00113
00114   command = find_command (word);
00115
00116   if (!command) {
00117     std::cerr << word << ": No such command for airinv." << std::endl;
00118     return -1;
00119   }
00120
00121   /* Get argument to command, if any. */
00122   while (whitespace (line[i])) {
00123     i++;
00124   }
00125
00126   word = line + i;
00127
00128   /* Call the function. */
00129   return (*(command->func)) (word);
00130 }
00131
00136 COMMAND* find_command (char* name) {
00137   register int i;
00138
00139   for (i = 0; commands[i].name; i++) {
00140     if (strcmp (name, commands[i].name) == 0) {
00141       return (&commands[i]);
00142     }
00143   }
00144
00145   return (COMMAND*) NULL;
00146 }
00147
00152 char* stripwhite (char* string) {
00153   register char *s, *t;
00154
00155   for (s = string; whitespace (*s); s++) {
00156   }
00157
00158   if (*s == 0) {
00159     return s;
00160   }
00161
00162   t = s + strlen (s) - 1;
```

```
00163   while (t > s && whitespace (*t)) {
00164     t--;
00165   }
00166   *++t = '\0';
00167 
00168   return s;
00169 }
00170 
00171 /* ********************************************************************** */
00172 /*                                                                        */
00173 /*                  Interface to Readline Completion                      */
00174 /*                                                                        */
00175 /* ********************************************************************** */
00176 
00177 char* command_generator (char* text, int state);
00178 char** fileman_completion (char* text, int start, int end);
00179 
00185 void initialize_readline() {
00186   /* Allow conditional parsing of the ~/.inputrc file. */
00187   rl_readline_name = "airinv";
00188 
00189   /* Tell the completer that we want a crack first. */
00190   rl_attempted_completion_function = (rl_completion_func_t*) fileman_completion
      ;
00191 }
00192 
00200 char** fileman_completion (char* text, int start, int end) {
00201   char **matches;
00202 
00203   matches = (char**) NULL;
00204 
00210   if (start == 0) {
00211     matches = completion_matches (text, command_generator);
00212   }
00213 
00214   return matches;
00215 }
00216 
00222 char* command_generator (char* text, int state) {
00223   static int list_index, len;
00224   char* name;
00225 
00231   if (!state) {
00232     list_index = 0;
00233     len = strlen (text);
00234   }
00235 
00236   /* Return the next name which partially matches from the command list. */
00237   while (name = commands[list_index].name) {
00238     ++list_index;
00239 
00240     if (strncmp (name, text, len) == 0) {
00241       return dupstr (name);
00242     }
00243   }
00244 
00245   /* If no names matched, then return NULL. */
00246   return (char*) NULL;
00247 }
00248 
00249 /* ********************************************************************** */
00250 /*                                                                        */
00251 /*                        airinv Commands                                 */
00252 /*                                                                        */
00253 /* ********************************************************************** */
00254 
00259 static char syscom[1024];
00260 
00264 void com_list (char* arg) {
00265   if (!arg) {
00266     arg = "";
00267   }
00268 
00269   std::ostringstream oStr;
```

```
00270   oStr << "ls -FClg " << arg;
00271   return system (oStr.c_str());
00272 }
00273
00274 int com_view (char* arg) {
00275   if (!valid_argument ("view", arg)) {
00276     return 1;
00277   }
00278
00279   std::ostringstream oStr;
00280   oStr << "more " << arg;
00281   return system (syscom);
00282 }
00283
00284 int com_rename (char* arg) {
00285   too_dangerous ("rename");
00286   return 1;
00287 }
00288
00289 int com_stat (char* arg) {
00290   struct stat finfo;
00291
00292   if (!valid_argument ("stat", arg)) {
00293     return 1;
00294   }
00295
00296   if (stat (arg, &finfo) == -1) {
00297     perror (arg);
00298     return 1;
00299   }
00300
00301   std::cout << "Statistics for `" << arg << "':" << std::endl;
00302
00303   const std::string lPluralEnd1 = (finfo.st_nlink == 1) ? "" : "s";
00304   const std::string lPluralEnd2 = (finfo.st_size == 1) ? "" : "s";
00305   std::cout << arg << " has "
00306             << finfo.st_nlink << " link" << lPluralEnd1 << ", and is "
00307             << finfo.st_size << " byte" << lPluralEnd2 << " in length."
00308             << std::endl;
00309   std::cout << " Inode Last Change at: " << ctime (&finfo.st_ctime) << std
      ::endl;
00310   std::cout << " Last access at: " << ctime (&finfo.st_atime) << std::endl;
00311   std::cout << " Last modified at: " << ctime (&finfo.st_mtime) << std::endl;
00312   return 0;
00313 }
00314
00315 int com_delete (char* arg) {
00316   too_dangerous ("delete");
00317   return 1;
00318 }
00319
00324 int com_help (char* arg) {
00325   register int i;
00326   int printed = 0;
00327
00328   for (i = 0; commands[i].name; i++) {
00329     if (!*arg || (strcmp (arg, commands[i].name) == 0)) {
00330       printf ("%s\t\t%s.\n", commands[i].name, commands[i].doc);
00331       printed++;
00332     }
00333   }
00334
00335   if (!printed) {
00336     printf ("No commands match `%s'.  Possibilties are:\n", arg);
00337
00338     for (i = 0; commands[i].name; i++) {
00339       /* Print in six columns. */
00340       if (printed == 6) {
00341         printed = 0;
00342         printf ("\n");
00343       }
00344
00345       printf ("%s\t", commands[i].name);
00346       printed++;
```

```
00347     }
00348
00349     if (printed)
00350       printf ("\n");
00351   }
00352   return 0;
00353 }
00354
00355 /* Change to the directory ARG. */
00356 int com_cd (char* arg) {
00357   if (chdir (arg) == -1) {
00358     perror (arg);
00359     return 1;
00360   }
00361
00362   com_pwd ("");
00363   return 0;
00364 }
00365
00366 /* Print out the current working directory. */
00367 int com_pwd (char* ignore) {
00368   char dir[1024], *s;
00369
00370   s = getwd (dir);
00371   if (s == 0) {
00372     printf ("Error getting pwd: %s\n", dir);
00373     return 1;
00374   }
00375
00376   printf ("Current directory is %s\n", dir);
00377   return 0;
00378 }
00379
00380 /* The user wishes to quit using this program.  Just set DONE non-zero. */
00381 int com_quit (char* arg) {
00382   done = 1;
00383   return 0;
00384 }
00385
00386 /* Function which tells you that you can't do this. */
00387 void too_dangerous (char* caller) {
00388   fprintf (stderr,
00389            "%s: Too dangerous for me to distribute.  Write it yourself.\n",
00390            caller);
00391 }
00392
00393 /* Return non-zero if ARG is a valid argument for CALLER, else print
00394  *    an error message and return zero. */
00395 int valid_argument (char* caller, char* arg) {
00396   if (!arg || !*arg) {
00397     fprintf (stderr, "%s: Argument required.\n", caller);
00398     return 0;
00399   }
00400
00401   return 1;
00402 }
00403
00404 #endif // _AIRINV_READLINE_AUTOCOMP_HPP
```

25.53 sevmgr/ui/cmdline/sevmgr.cpp File Reference

25.54 sevmgr.cpp

```
00001
00005 // STL
00006 #include <cassert>
00007 #include <iostream>
00008 #include <sstream>
00009 #include <fstream>
00010 #include <string>
```

```
00011 // Boost (Extended STL)
00012 #include <boost/program_options.hpp>
00013 #include <boost/tokenizer.hpp>
00014 #include <boost/regex.hpp>
00015 #include <boost/swap.hpp>
00016 #include <boost/algorithm/string/case_conv.hpp>
00017 // StdAir
00018 #include <stdair/basic/BasLogParams.hpp>
00019 #include <stdair/basic/BasDBParams.hpp>
00020 #include <stdair/service/Logger.hpp>
00021 // SEvMgr
00022 #include <sevmgr/SEVMGR_Service.hpp>
00023 #include <sevmgr/config/sevmgr-paths.hpp>
00024 // GNU Readline Wrapper
00025 #include <sevmgr/ui/cmdline/SReadline.hpp>
00026
00027 // //////// Constants //////
00031 const std::string K_SEVMGR_DEFAULT_LOG_FILENAME ("sevmgr.log");
00032
00036 const int K_SEVMGR_EARLY_RETURN_STATUS = 99;
00037
00042 typedef std::vector<std::string> TokenList_T;
00043
00047 struct Command_T {
00048   typedef enum {
00049     NOP = 0,
00050     QUIT,
00051     HELP,
00052     LIST,
00053     DISPLAY,
00054     SELECT,
00055     NEXT,
00056     RUN,
00057     JSON_LIST,
00058     JSON_DISPLAY,
00059     LAST_VALUE
00060   } Type_T;
00061 };
00062
00063 // //////// Parsing of Options & Configuration ////////
00064 // A helper function to simplify the main part.
00065 template<class T> std::ostream& operator<< (std::ostream& os,
00066                                             const std::vector<T>& v) {
00067   std::copy (v.begin(), v.end(), std::ostream_iterator<T> (std::cout, " "));
00068   return os;
00069 }
00070
00074 int readConfiguration (int argc, char* argv[], std::string& ioLogFilename) {
00075   // Declare a group of options that will be allowed only on command line
00076   boost::program_options::options_description generic ("Generic options");
00077   generic.add_options()
00078     ("prefix", "print installation prefix")
00079     ("version,v", "print version string")
00080     ("help,h", "produce help message");
00081
00082   // Declare a group of options that will be allowed both on command
00083   // line and in config file
00084
00085   boost::program_options::options_description config ("Configuration");
00086   config.add_options()
00087     ("log,l",
00088      boost::program_options::value< std::string >(&ioLogFilename)->
      default_value(K_SEVMGR_DEFAULT_LOG_FILENAME),
00089      "Filename for the logs")
00090     ;
00091
00092   // Hidden options, will be allowed both on command line and
00093   // in config file, but will not be shown to the user.
00094   boost::program_options::options_description hidden ("Hidden options");
00095   hidden.add_options()
00096     ("copyright",
00097      boost::program_options::value< std::vector<std::string> >(),
00098      "Show the copyright (license)");
00099
```

```
00100   boost::program_options::options_description cmdline_options;
00101   cmdline_options.add(generic).add(config).add(hidden);
00102
00103   boost::program_options::options_description config_file_options;
00104   config_file_options.add(config).add(hidden);
00105   boost::program_options::options_description visible ("Allowed options");
00106   visible.add(generic).add(config);
00107
00108   boost::program_options::positional_options_description p;
00109   p.add ("copyright", -1);
00110
00111   boost::program_options::variables_map vm;
00112   boost::program_options::
00113     store (boost::program_options::command_line_parser (argc, argv).
00114            options (cmdline_options).positional(p).run(), vm);
00115
00116   std::ifstream ifs ("sevmgr.cfg");
00117   boost::program_options::store (parse_config_file (ifs, config_file_options),
00118                                  vm);
00119   boost::program_options::notify (vm);
00120
00121   if (vm.count ("help")) {
00122     std::cout << visible << std::endl;
00123     return K_SEVMGR_EARLY_RETURN_STATUS;
00124   }
00125
00126   if (vm.count ("version")) {
00127     std::cout << PACKAGE_NAME << ", version " << PACKAGE_VERSION << std::endl;
00128     return K_SEVMGR_EARLY_RETURN_STATUS;
00129   }
00130
00131   if (vm.count ("prefix")) {
00132     std::cout << "Installation prefix: " << PREFIXDIR << std::endl;
00133     return K_SEVMGR_EARLY_RETURN_STATUS;
00134   }
00135
00136   if (vm.count ("log")) {
00137     ioLogFilename = vm["log"].as< std::string >();
00138     std::cout << "Log filename is: " << ioLogFilename << std::endl;
00139   }
00140
00141   return 0;
00142 }
00143
00144 // ////////////////////////////////////////////////////////////////////
00145 void initReadline (swift::SReadline& ioInputReader) {
00146
00147   // Prepare the list of my own completers
00148   std::vector<std::string> Completers;
00149
00150   // The following is supported:
00151   // - "identifiers"
00152   // - special identifier %file - means to perform a file name completion
00153   Completers.push_back ("help");
00154   Completers.push_back ("list %airline_code %flight_number");
00155   Completers.push_back ("select %airline_code %flight_number %flight_date");
00156   Completers.push_back ("display");
00157   Completers.push_back ("next");
00158   Completers.push_back ("run");
00159   Completers.push_back ("quit");
00160
00161
00162   // Now register the completers.
00163   // Actually it is possible to re-register another set at any time
00164   ioInputReader.RegisterCompletions (Completers);
00165 }
00166
00167 // ////////////////////////////////////////////////////////////////////
00168 Command_T::Type_T extractCommand (TokenList_T& ioTokenList) {
00169   Command_T::Type_T oCommandType = Command_T::LAST_VALUE;
00170
00171   // Interpret the user input
00172   if (ioTokenList.empty() == false) {
00173     TokenList_T::iterator itTok = ioTokenList.begin();
```

```
00174     std::string lCommand (*itTok);
00175     boost::algorithm::to_lower (lCommand);
00176
00177     if (lCommand == "help") {
00178       oCommandType = Command_T::HELP;
00179
00180     } else if (lCommand == "list") {
00181       oCommandType = Command_T::LIST;
00182
00183     } else if (lCommand == "display") {
00184       oCommandType = Command_T::DISPLAY;
00185
00186     } else if (lCommand == "select") {
00187       oCommandType = Command_T::SELECT;
00188
00189     } else if (lCommand == "next") {
00190       oCommandType = Command_T::NEXT;
00191
00192     } else if (lCommand == "run") {
00193       oCommandType = Command_T::RUN;
00194
00195     } else if (lCommand == "json_list") {
00196       oCommandType = Command_T::JSON_LIST;
00197
00198     } else if (lCommand == "json_display") {
00199       oCommandType = Command_T::JSON_DISPLAY;
00200
00201     } else if (lCommand == "quit") {
00202       oCommandType = Command_T::QUIT;
00203     }
00204
00205     // Remove the first token (the command), as the corresponding information
00206     // has been extracted in the form of the returned command type enumeration
00207     ioTokenList.erase (itTok);
00208
00209   } else {
00210     oCommandType = Command_T::NOP;
00211   }
00212
00213   return oCommandType;
00214 }
00215
00216 // //////////////////////////////////////////////////////////////////
00217 void parseFlightKey (const TokenList_T& iTokenList,
00218                      stdair::AirlineCode_T& ioAirlineCode,
00219                      stdair::FlightNumber_T& ioFlightNumber) {
00220   // Interpret the user input
00221   if (iTokenList.empty() == false) {
00222
00223     // Read the airline code
00224     TokenList_T::const_iterator itTok = iTokenList.begin();
00225     if (itTok->empty() == false) {
00226       ioAirlineCode = *itTok;
00227       boost::algorithm::to_upper (ioAirlineCode);
00228     }
00229
00230     // Read the flight-number
00231     ++itTok;
00232     if (itTok != iTokenList.end()) {
00233
00234       if (itTok->empty() == false) {
00235         try {
00236
00237           ioFlightNumber = boost::lexical_cast<stdair::FlightNumber_T> (*itTok)
      ;
00238
00239         } catch (boost::bad_lexical_cast& eCast) {
00240           std::cerr << "The flight number ('" << *itTok
00241                     << "') cannot be understood. "
00242                     << "The default value (all) is kept."
00243                     << std::endl;
00244           return;
00245         }
00246       }
```

```
00247
00248     } else {
00249       return;
00250     }
00251   }
00252 }
00253
00254 // ////////////////////////////////////////////////////////////////////
00255 void parseFlightDateKey (const TokenList_T& iTokenList,
00256                          stdair::AirlineCode_T& ioAirlineCode,
00257                          stdair::FlightNumber_T& ioFlightNumber,
00258                          stdair::Date_T& ioDepartureDate) {
00259   //
00260   const std::string kMonthStr[12] = {"Jan", "Feb", "Mar", "Apr", "May", "Jun",
00261                                      "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"};
00262   //
00263   unsigned short ioDepartureDateYear = ioDepartureDate.year();
00264   unsigned short ioDepartureDateMonth = ioDepartureDate.month();
00265   std::string ioDepartureDateMonthStr = kMonthStr[ioDepartureDateMonth-1];
00266   unsigned short ioDepartureDateDay = ioDepartureDate.day();
00267
00268   // Interpret the user input
00269   if (iTokenList.empty() == false) {
00270
00271     // Read the airline code
00272     TokenList_T::const_iterator itTok = iTokenList.begin();
00273     if (itTok->empty() == false) {
00274       ioAirlineCode = *itTok;
00275       boost::algorithm::to_upper (ioAirlineCode);
00276     }
00277
00278     // Read the flight-number
00279     ++itTok;
00280     if (itTok != iTokenList.end()) {
00281
00282       if (itTok->empty() == false) {
00283         try {
00284
00285           ioFlightNumber = boost::lexical_cast<stdair::FlightNumber_T> (*itTok)
      ;
00286
00287         } catch (boost::bad_lexical_cast& eCast) {
00288           std::cerr << "The flight number ('" << *itTok
00289                     << "') cannot be understood. "
00290                     << "The default value (all) is kept."
00291                     << std::endl;
00292           return;
00293         }
00294       }
00295
00296     } else {
00297       return;
00298     }
00299
00300     // Read the year for the departure date
00301     ++itTok;
00302     if (itTok != iTokenList.end()) {
00303
00304       if (itTok->empty() == false) {
00305         try {
00306
00307           ioDepartureDateYear = boost::lexical_cast<unsigned short> (*itTok);
00308           if (ioDepartureDateYear < 100) {
00309             ioDepartureDateYear += 2000;
00310           }
00311
00312         } catch (boost::bad_lexical_cast& eCast) {
00313           std::cerr << "The year of the flight departure date ('" << *itTok
00314                     << "') cannot be understood. The default value ("
00315                     << ioDepartureDateYear << ") is kept. " << std::endl;
00316           return;
00317         }
00318       }
00319
```

```
00320     } else {
00321       return;
00322     }
00323
00324     // Read the month for the departure date
00325     ++itTok;
00326     if (itTok != iTokenList.end()) {
00327
00328       if (itTok->empty() == false) {
00329         try {
00330
00331           const boost::regex lMonthRegex ("^(\\d{1,2})$");
00332           const bool isMonthANumber = regex_match (*itTok, lMonthRegex);
00333
00334           if (isMonthANumber == true) {
00335             const unsigned short lMonth =
00336               boost::lexical_cast<unsigned short> (*itTok);
00337             if (lMonth > 12) {
00338               throw boost::bad_lexical_cast();
00339             }
00340             ioDepartureDateMonthStr = kMonthStr[lMonth-1];
00341
00342           } else {
00343             const std::string lMonthStr (*itTok);
00344             if (lMonthStr.size() < 3) {
00345               throw boost::bad_lexical_cast();
00346             }
00347             std::string lMonthStr1 (lMonthStr.substr (0, 1));
00348             boost::algorithm::to_upper (lMonthStr1);
00349             std::string lMonthStr23 (lMonthStr.substr (1, 2));
00350             boost::algorithm::to_lower (lMonthStr23);
00351             ioDepartureDateMonthStr = lMonthStr1 + lMonthStr23;
00352           }
00353
00354         } catch (boost::bad_lexical_cast& eCast) {
00355           std::cerr << "The month of the flight departure date ('" << *itTok
00356                     << "') cannot be understood. The default value ("
00357                     << ioDepartureDateMonthStr << ") is kept. " << std::endl;
00358           return;
00359         }
00360       }
00361
00362     } else {
00363       return;
00364     }
00365
00366     // Read the day for the departure date
00367     ++itTok;
00368     if (itTok != iTokenList.end()) {
00369
00370       if (itTok->empty() == false) {
00371         try {
00372
00373           ioDepartureDateDay = boost::lexical_cast<unsigned short> (*itTok);
00374
00375         } catch (boost::bad_lexical_cast& eCast) {
00376           std::cerr << "The day of the flight departure date ('" << *itTok
00377                     << "') cannot be understood. The default value ("
00378                     << ioDepartureDateDay << ") is kept. " << std::endl;
00379           return;
00380         }
00381       }
00382
00383     } else {
00384       return;
00385     }
00386
00387     // Re-compose the departure date
00388     std::ostringstream lDepartureDateStr;
00389     lDepartureDateStr << ioDepartureDateYear << "-" << ioDepartureDateMonthStr
00390                       << "-" << ioDepartureDateDay;
00391
00392     try {
00393
```

```
00394       ioDepartureDate =
00395         boost::gregorian::from_simple_string (lDepartureDateStr.str());
00396
00397     } catch (boost::gregorian::bad_month& eCast) {
00398       std::cerr << "The flight departure date ('" << lDepartureDateStr.str()
00399                 << "') cannot be understood. The default value ("
00400                 << ioDepartureDate << ") is kept. " << std::endl;
00401       return;
00402     }
00403
00404   }
00405 }
00406
00407 // ////////////////////////////////////////////////////////////////////
00408 void parseBookingClassKey (const TokenList_T& iTokenList,
00409                            stdair::ClassCode_T& ioBookingClass,
00410                            stdair::PartySize_T& ioPartySize,
00411                            stdair::AirportCode_T& ioOrigin,
00412                            stdair::AirportCode_T& ioDestination) {
00413   // Interpret the user input
00414   if (iTokenList.empty() == false) {
00415
00416     // Read the booking class
00417     TokenList_T::const_iterator itTok = iTokenList.begin();
00418     if (itTok->empty() == false) {
00419       ioBookingClass = *itTok;
00420       boost::algorithm::to_upper (ioBookingClass);
00421     }
00422
00423     // Read the party size
00424     ++itTok;
00425     if (itTok != iTokenList.end()) {
00426
00427       if (itTok->empty() == false) {
00428         try {
00429
00430           ioPartySize = boost::lexical_cast<stdair::PartySize_T> (*itTok);
00431
00432         } catch (boost::bad_lexical_cast& eCast) {
00433           std::cerr << "The party size ('" << *itTok
00434                     << "') cannot be understood. The default value ("
00435                     << ioPartySize << ") is kept." << std::endl;
00436           return;
00437         }
00438       }
00439
00440     } else {
00441       return;
00442     }
00443
00444     // Read the origin
00445     ++itTok;
00446     if (itTok != iTokenList.end()) {
00447
00448       if (itTok->empty() == false) {
00449         ioOrigin = *itTok;
00450         boost::algorithm::to_upper (ioOrigin);
00451       }
00452
00453     } else {
00454       return;
00455     }
00456
00457     // Read the destination
00458     ++itTok;
00459     if (itTok != iTokenList.end()) {
00460
00461       if (itTok->empty() == false) {
00462         ioDestination = *itTok;
00463         boost::algorithm::to_upper (ioDestination);
00464       }
00465
00466     } else {
00467       return;
```

```
00468     }
00469   }
00470 }
00471
00472 // ////////////////////////////////////////////////////////////
00473 std::string toString (const TokenList_T& iTokenList) {
00474   std::ostringstream oStr;
00475
00476   // Re-create the string with all the tokens, trimmed by read-line
00477   unsigned short idx = 0;
00478   for (TokenList_T::const_iterator itTok = iTokenList.begin();
00479        itTok != iTokenList.end(); ++itTok, ++idx) {
00480     if (idx != 0) {
00481       oStr << " ";
00482     }
00483     oStr << *itTok;
00484   }
00485
00486   return oStr.str();
00487 }
00488
00489 // ////////////////////////////////////////////////////////////
00490 TokenList_T extractTokenList (const TokenList_T& iTokenList,
00491                               const std::string& iRegularExpression) {
00492   TokenList_T oTokenList;
00493
00494   // Re-create the string with all the tokens (which had been trimmed
00495   // by read-line)
00496   const std::string lFullLine = toString (iTokenList);
00497
00498   // See the caller for the regular expression
00499   boost::regex expression (iRegularExpression);
00500
00501   std::string::const_iterator start = lFullLine.begin();
00502   std::string::const_iterator end = lFullLine.end();
00503
00504   boost::match_results<std::string::const_iterator> what;
00505   boost::match_flag_type flags = boost::match_default | boost::format_sed;
00506   regex_search (start, end, what, expression, flags);
00507
00508   // Put the matched strings in the list of tokens to be returned back
00509   // to the caller
00510   const unsigned short lMatchSetSize = what.size();
00511   for (unsigned short matchIdx = 1; matchIdx != lMatchSetSize; ++matchIdx) {
00512     const std::string lMatchedString (std::string (what[matchIdx].first,
00513                                                    what[matchIdx].second));
00514     //if (lMatchedString.empty() == false) {
00515     oTokenList.push_back (lMatchedString);
00516     //}
00517   }
00518
00519   // DEBUG
00520   // std::cout << "After (token list): " << oTokenList << std::endl;
00521
00522   return oTokenList;
00523 }
00524
00525 // ////////////////////////////////////////////////////////////
00526 TokenList_T extractTokenListForFlight (const TokenList_T& iTokenList) {
00533   const std::string lRegEx ("^([[:alpha:]]{2,3})?"
00534                             "[[:space:]]*([[:digit:]]{1,4})?$");
00535
00536   //
00537   const TokenList_T& oTokenList = extractTokenList (iTokenList, lRegEx);
00538   return oTokenList;
00539 }
00540
00541 // ////////////////////////////////////////////////////////////
00542 TokenList_T extractTokenListForFlightDate (const TokenList_T& iTokenList) {
00553   const std::string lRegEx("^([[:alpha:]]{2,3})?"
00554                            "[[:space:]]*([[:digit:]]{1,4})?"
00555                            "[/ ]*"
00556                            "([[:digit:]]{2,4})?[/-]?[[:space:]]*"
00557                            "([[:alpha:]]{3}|[[:digit:]]{1,2})?[/-]?[[:space:]]*
```

```
    "
00558                           "([[:digit:]]{1,2})?$");
00559
00560   //
00561   const TokenList_T& oTokenList = extractTokenList (iTokenList, lRegEx);
00562   return oTokenList;
00563 }
00564
00565 // ////////////////////////////////////////////////////
00566 TokenList_T extractTokenListForClass (const TokenList_T& iTokenList) {
00575   const std::string lRegEx ("^([[:alpha:]])?"
00576                             "[[:space:]]*([[:digit:]]{1,3})?"
00577                             "[[:space:]]*([[:alpha:]]{3})?"
00578                             "[[:space:]]*([[:alpha:]]{3})?$");
00579
00580   //
00581   const TokenList_T& oTokenList = extractTokenList (iTokenList, lRegEx);
00582   return oTokenList;
00583 }
00584
00585
00586 // ///////// M A I N /////////////
00587 int main (int argc, char* argv[]) {
00588
00589   // Readline history
00590   const unsigned int lHistorySize (100);
00591   const std::string lHistoryFilename ("sevmgr.hist");
00592   const std::string lHistoryBackupFilename ("sevmgr.hist.bak");
00593
00594   // Default parameters for the interactive session
00595   stdair::AirlineCode_T lLastInteractiveAirlineCode;
00596   stdair::FlightNumber_T lLastInteractiveFlightNumber;
00597   stdair::Date_T lLastInteractiveDate;
00598   stdair::AirlineCode_T lInteractiveAirlineCode;
00599   stdair::FlightNumber_T lInteractiveFlightNumber;
00600   stdair::Date_T lInteractiveDate;
00601   stdair::AirportCode_T lInteractiveOrigin;
00602   stdair::AirportCode_T lInteractiveDestination;
00603   stdair::ClassCode_T lInteractiveBookingClass;
00604
00605   // Parameters for the sale
00606   std::string lSegmentDateKey;
00607
00608   // Output log File
00609   stdair::Filename_T lLogFilename;
00610
00611   // Call the command-line option parser
00612   const int lOptionParserStatus = readConfiguration (argc, argv, lLogFilename);
00613
00614   if (lOptionParserStatus == K_SEVMGR_EARLY_RETURN_STATUS) {
00615     return 0;
00616   }
00617
00618   // Set the log parameters
00619   std::ofstream logOutputFile;
00620   // Open and clean the log outputfile
00621   logOutputFile.open (lLogFilename.c_str());
00622   logOutputFile.clear();
00623
00624   // Initialise the inventory service
00625   const stdair::BasLogParams lLogParams (stdair::LOG::DEBUG, logOutputFile);
00626   SEVMGR::SEVMGR_Service sevmgrService (lLogParams);
00627
00628   // DEBUG
00629   STDAIR_LOG_DEBUG ("Welcome to SEvMgr");
00630
00631   // Build the sample BOM tree for RMOL
00632   sevmgrService.buildSampleBom();
00633
00634   // Update the default parameters for the following interactive session
00635   lInteractiveAirlineCode = "BA";
00636   lInteractiveFlightNumber = 9;
00637   lInteractiveDate = stdair::Date_T (2011, 06, 10);
00638   lInteractiveBookingClass = "Q";
```

```
00639   lInteractiveOrigin = "LHR";
00640   lInteractiveDestination = "SYD";
00641
00642   // Save the last state
00643   lLastInteractiveAirlineCode = lInteractiveAirlineCode;
00644   lLastInteractiveFlightNumber = lInteractiveFlightNumber;
00645   lLastInteractiveDate = lInteractiveDate;
00646
00647   // DEBUG
00648   STDAIR_LOG_DEBUG ("=================================================");
00649   STDAIR_LOG_DEBUG ("=       Beginning of the interactive session      =");
00650   STDAIR_LOG_DEBUG ("=================================================");
00651   STDAIR_LOG_DEBUG ("Last saved state: " << lLastInteractiveAirlineCode
00652                     << lLastInteractiveFlightNumber << " / "
00653                     << lLastInteractiveDate);
00654
00655   // Initialise the GNU readline wrapper
00656   swift::SReadline lReader (lHistoryFilename, lHistorySize);
00657   initReadline (lReader);
00658
00659   // Now we can ask user for a line
00660   std::string lUserInput;
00661   bool EndOfInput (false);
00662   Command_T::Type_T lCommandType (Command_T::NOP);
00663
00664   while (lCommandType != Command_T::QUIT && EndOfInput == false) {
00665     // Prompt
00666     std::ostringstream oPromptStr;
00667     oPromptStr << "sevmgr "
00668                << lInteractiveAirlineCode << lInteractiveFlightNumber
00669                << " / " << lInteractiveDate
00670                << "> ";
00671     // Call read-line, which will fill the list of tokens
00672     TokenList_T lTokenListByReadline;
00673     lUserInput = lReader.GetLine (oPromptStr.str(), lTokenListByReadline,
00674                                   EndOfInput);
00675
00676     // The history can be saved to an arbitrary file at any time
00677     lReader.SaveHistory (lHistoryBackupFilename);
00678
00679     // The end-of-input typically corresponds to a CTRL-D typed by the user
00680     if (EndOfInput) {
00681       std::cout << std::endl;
00682       break;
00683     }
00684
00685     // Interpret the user input
00686     lCommandType = extractCommand (lTokenListByReadline);
00687
00688     switch (lCommandType) {
00689
00690       // ////////////////////////////// Help ////////////////////////
00691     case Command_T::HELP: {
00692       std::cout << std::endl;
00693       std::cout << "Commands: " << std::endl;
00694       std::cout << " help" << "\t\t" << "Display this help" << std::endl;
00695       std::cout << " quit" << "\t\t" << "Quit the application" << std::endl;
00696       std::cout << " list" << "\t\t" << "List events" << std::endl;
00697       std::cout << " select" << "\t\t"
00698                 << "Select an event to become the current one" << std::endl;
00699       std::cout << " display" << "\t"
00700                 << "Display the current event" << std::endl;
00701       std::cout << " next" << "\t\t"
00702                 << "Play the current event and pop the next one from the queue"
00703                 << std::endl;
00704       std::cout << " run" << "\t\t"
00705                 << "Play all the events until the next break-point, if any"
00706                 << std::endl;
00707       std::cout << " \nDebug Commands" << std::endl;
00708       std::cout << " json_list" << "\t"
00709                 << "List events in a JSON format"
00710                 << std::endl;
00711       std::cout << " json_display" << "\t"
00712                 << "Display the current event in a JSON format"
```

```
00713                 << std::endl;
00714       std::cout << std::endl;
00715       break;
00716     }
00717
00718       // ////////////////////////////// Quit //////////////////////
00719     case Command_T::QUIT: {
00720       break;
00721     }
00722
00723       // ////////////////////////////// List //////////////////////
00724     case Command_T::LIST: {
00725       //
00726       std::cout << "List" << std::endl;
00727
00728       //
00729       break;
00730     }
00731
00732       // ////////////////////////////// Select //////////////////////
00733     case Command_T::SELECT: {
00734       //
00735       std::cout << "Select" << std::endl;
00736
00737       //
00738       break;
00739     }
00740
00741       // ////////////////////////////// Display //////////////////////
00742     case Command_T::DISPLAY: {
00743       //
00744       std::cout << "Display" << std::endl;
00745
00746       //
00747       break;
00748     }
00749
00750       // ////////////////////////////// Next //////////////////////
00751     case Command_T::NEXT: {
00752       //
00753       std::cout << "Next" << std::endl;
00754
00755       //
00756       break;
00757     }
00758
00759       // ////////////////////////////// Run //////////////////////
00760     case Command_T::RUN: {
00761       //
00762       std::cout << "Run" << std::endl;
00763
00764       //
00765       break;
00766     }
00767
00768       // ////////////////////////////// JSon List //////////////////////
00769
00770     case Command_T::JSON_LIST: {
00771       //
00772       std::cout << "JSON List" << std::endl;
00773
00774       //
00775       break;
00776     }
00777
00778       // ////////////////////////////// JSon Display //////////////////////
00779
00780     case Command_T::JSON_DISPLAY: {
00781       //
00782       std::cout << "JSON Display" << std::endl;
00783
00784       //
00785       break;
00786     }
```

```
00787
00788       // ////////////////////////// Default / No value ///////////////////////
00789     case Command_T::NOP: {
00790       break;
00791     }
00792
00793     case Command_T::LAST_VALUE:
00794     default: {
00795       // DEBUG
00796       std::ostringstream oStr;
00797       oStr << "That command is not yet understood: '" << lUserInput
00798            << "' => " << lTokenListByReadline;
00799       STDAIR_LOG_DEBUG (oStr.str());
00800       std::cout << oStr.str() << std::endl;
00801     }
00802     }
00803   }
00804
00805   // DEBUG
00806   STDAIR_LOG_DEBUG ("End of the session. Exiting.");
00807   std::cout << "End of the session. Exiting." << std::endl;
00808
00809   // Close the Log outputFile
00810   logOutputFile.close();
00811
00812   /*
00813     Note: as that program is not intended to be run on a server in
00814     production, it is better not to catch the exceptions. When it
00815     happens (that an exception is thrownd), that way we get the
00816     call stack.
00817   */
00818
00819   return 0;
00820 }
```

25.55 sevmgr/ui/cmdline/SReadline.hpp File Reference

C++ wrapper around libreadline.

`#include <cstdio> #include <readline/readline.h> #include <readline/history.h> #include <readline/keymaps.h> × #include <string> #include <fstream> #include <vector> #include <stdexcept> #include <map> #include <boost/algorithm/string/trim.-hpp> #include <boost/tokenizer.hpp> #include <boost/function.-hpp>`

Classes

- class swift::SKeymap

  The readline keymap wrapper.
- class swift::SReadline

  The readline library wrapper.

Namespaces

- namespace swift

  The wrapper namespace.

25.55.1 Detailed Description

C++ wrapper around libreadline. Supported: editing, history, custom completers, keymaps. Attention: implementation is not thread safe! It is mainly because the readline library provides pure C interface and has many calls for an "atomic" completion operation

Definition in file SReadline.hpp.

25.56 SReadline.hpp

```
00001
00011 //
00012 // Date:      17 December 2005
00013 //            03 April    2006
00014 //            20 April    2006
00015 //            07 May      2006
00016 //
00017 // Copyright (c) Sergey Satskiy 2005 - 2006
00018 //               <sergesatsky@yahoo.com>
00019 //
00020 // Permission to copy, use, modify, sell and distribute this software
00021 // is granted provided this copyright notice appears in all copies.
00022 // This software is provided "as is" without express or implied
00023 // warranty, and with no claim as to its suitability for any purpose.
00024 //
00025
00026 #ifndef SREADLINE_H
00027 #define SREADLINE_H
00028
00029 #include <cstdio>
00030
00031 #include <readline/readline.h>
00032 #include <readline/history.h>
00033 #include <readline/keymaps.h>
00034
00035 #include <string>
00036 #include <fstream>
00037 #include <vector>
00038 #include <stdexcept>
00039 #include <map>
00040
00041 #include <boost/algorithm/string/trim.hpp>
00042 #include <boost/tokenizer.hpp>
00043 #include <boost/function.hpp>
00044
00045
00050 namespace {
00054   typedef std::vector<std::string> TokensStorage;
00055
00059   typedef std::vector<TokensStorage> CompletionsStorage;
00060
00064   typedef boost::function<int (int, int)> KeyCallback;
00065
00069   typedef std::map<int, KeyCallback> KeysBind;
00070
00074   const size_t DefaultHistoryLimit (64);
00075
00079   CompletionsStorage Completions;
00080
00084   TokensStorage Tokens;
00085
00089   std::map<Keymap, KeysBind> Keymaps;
00090
00094   bool KeymapWasSetup (false);
00095
00099   Keymap Earlykeymap (0);
00100
```

```
00101
00108   char* Generator (const char* text, int State);
00109
00110
00118   char** UserCompletion (const char* text, int start, int end);
00119
00120
00128   int KeyDispatcher (int Count, int Key);
00129
00130
00135   int StartupHook (void);
00136
00137
00145   template <typename  Container>
00146   bool AreTokensEqual (const Container& Pattern, const Container& Input) {
00147     if (Input.size() > Pattern.size()) {
00148       return false;
00149     }
00150
00151     typename Container::const_iterator k (Pattern.begin());
00152     typename Container::const_iterator j (Input.begin());
00153     for ( ; j != Input.end(); ++k, ++j) {
00154       const std::string lPattern = *k;
00155       if (lPattern == "%file") {
00156         continue;
00157       }
00158
00159       const std::string lInput = *j;
00160       if (lPattern != lInput) {
00161         return false;
00162       }
00163     }
00164     return true;
00165   }
00166
00167   // See description near the prototype
00168   template <typename  ContainerType>
00169   void SplitTokens (const std::string& Source, ContainerType& Container) {
00170     typedef boost::tokenizer<boost::char_separator<char> > TokenizerType;
00171
00172     // Set of token separators
00173     boost::char_separator<char> Separators (" \t\n");
00174     // Tokens provider
00175     TokenizerType Tokenizer (Source, Separators);
00176
00177     Container.clear();
00178     for (TokenizerType::const_iterator k (Tokenizer.begin());
00179          k != Tokenizer.end(); ++k) {
00180       // Temporary storage for the token, in order to trim that latter
00181       std::string SingleToken (*k);
00182
00183       boost::algorithm::trim (SingleToken);
00184       Container.push_back (SingleToken);
00185     }
00186   }
00187
00188   // See description near the prototype
00189   char** UserCompletion (const char* text, int start, int end) {
00190     // No default completion at all
00191     rl_attempted_completion_over = 1;
00192
00193     if (Completions.empty() == true) {
00194       return NULL;
00195     }
00196
00197     // Memorise all the previous tokens
00198     std::string PreInput (rl_line_buffer, start);
00199     SplitTokens (PreInput, Tokens);
00200
00201     // Detect whether we should call the standard file name completer
00202     // or a custom one
00203     bool FoundPretender (false);
00204
00205     for (CompletionsStorage::const_iterator k (Completions.begin());
```

```
00206           k != Completions.end(); ++k) {
00207       const TokensStorage& lTokenStorage = *k;
00208       if (AreTokensEqual (lTokenStorage, Tokens) == false) {
00209         continue;
00210       }
00211
00212       if (lTokenStorage.size() > Tokens.size()) {
00213         FoundPretender = true;
00214         if (lTokenStorage [Tokens.size()] == "%file") {
00215           // Standard file name completer - called for the "%file" keyword
00216           return rl_completion_matches (text, rl_filename_completion_function);
00217         }
00218       }
00219     }
00220
00221     if (FoundPretender) {
00222       return rl_completion_matches (text, Generator);
00223     }
00224     return NULL;
00225   }
00226
00227   // See description near the prototype
00228   char* Generator (const char* text, int State) {
00229     static int Length;
00230     static CompletionsStorage::const_iterator Iterator;
00231
00232     if ( State == 0 ) {
00233       Iterator = Completions.begin();
00234       Length = strlen (text);
00235     }
00236
00237     for ( ; Iterator != Completions.end(); ++Iterator) {
00238       const TokensStorage& lCompletion = *Iterator;
00239       if (AreTokensEqual (lCompletion, Tokens) == false) {
00240         continue;
00241       }
00242
00243       if (lCompletion.size() > Tokens.size()) {
00244         if (lCompletion [Tokens.size()] == "%file") {
00245           continue;
00246         }
00247
00248         const char* lCompletionCharStr (lCompletion [Tokens.size()].c_str());
00249         if (strncmp (text, lCompletionCharStr, Length) == 0) {
00250           // Readline will free the allocated memory
00251           const size_t lCompletionSize = strlen (lCompletionCharStr) + 1;
00252           char* NewString (static_cast<char*> (malloc (lCompletionSize)));
00253           strcpy (NewString, lCompletionCharStr);
00254
00255           ++Iterator;
00256
00257           return NewString;
00258         }
00259       }
00260     }
00261
00262     return NULL;
00263   }
00264
00265
00266   // See the description near the prototype
00267   int KeyDispatcher  (int Count, int  Key ) {
00268     std::map< Keymap, KeysBind >::iterator Set (Keymaps.find (rl_get_keymap()))
    ;
00269     if (Set == Keymaps.end()) {
00270       // Most probably it happens bacause the header was
00271       // included into many compilation units and the
00272       // keymap setting calls were made in different files.
00273       // This is the problem of "global" data.
00274       // The storage of all the registered keymaps is in anonymous
00275       // namespace.
00276       throw std::runtime_error ("Error selecting a keymap.");
00277     }
00278
```

```
00279     (Set->second)[Key] (Count, Key);
00280     return 0;
00281   }
00282
00283   // See the description near the prototype
00284   int StartupHook (void) {
00285     if (KeymapWasSetup) {
00286       rl_set_keymap (Earlykeymap);
00287     }
00288     return 0;
00289   }
00290
00291 } // Anonymous namespace
00292
00293
00299 namespace swift {
00300
00307   class SKeymap {
00308   private:
00309     // Readline keymap
00310     Keymap keymap;
00311
00312   public:
00319     explicit SKeymap (bool PrintableBound = false) : keymap (NULL) {
00320       if (PrintableBound == true) {
00321         // Printable characters are bound
00322         keymap = rl_make_keymap();
00323
00324       } else {
00325         // Empty keymap
00326         keymap = rl_make_bare_keymap();
00327       }
00328
00329       if (keymap == NULL) {
00330         throw std::runtime_error ("Cannot allocate keymap.");
00331       }
00332
00333       // Register a new keymap in the global list
00334       Keymaps [keymap] = KeysBind();
00335     }
00336
00342     explicit SKeymap (Keymap Pattern) : keymap (rl_copy_keymap (Pattern)) {
00343       if ( keymap == NULL ) {
00344         throw std::runtime_error( "Cannot allocate keymap." );
00345       }
00346
00347       // Register a new keymap in the global list
00348       Keymaps [keymap] = KeysBind();
00349     }
00350
00354     ~SKeymap() {
00355       // Deregister the keymap
00356       Keymaps.erase (keymap);
00357       rl_discard_keymap (keymap);
00358     }
00359
00366     void Bind (int  Key, KeyCallback  Callback) {
00367       Keymaps [keymap][Key] = Callback;
00368
00369       if (rl_bind_key_in_map (Key, KeyDispatcher, keymap) != 0) {
00370         // Remove from the map just bound key
00371         Keymaps [keymap].erase (Key);
00372         throw std::runtime_error ("Invalid key.");
00373       }
00374     }
00375
00381     void Unbind (int  Key) {
00382       rl_unbind_key_in_map (Key, keymap);
00383       Keymaps [keymap].erase (Key);
00384     }
00385
00386     // void Bind (const std::string& Sequence, boost::function<int (int,
      int)>);
00387     // void Unbind (std::string&  Sequence);
```

```
00388
00389   public:
00395     SKeymap (const SKeymap& rhs) {
00396       if (this == &rhs) {
00397         return;
00398       }
00399       keymap = rl_copy_keymap (rhs.keymap);
00400     }
00401
00407     SKeymap& operator= (const SKeymap& rhs) {
00408       if (this == &rhs) {
00409         return *this;
00410       }
00411       keymap = rl_copy_keymap (rhs.keymap);
00412       return *this;
00413     }
00414
00415     friend class SReadline;
00416   };
00417
00424   class SReadline {
00425   public:
00431     SReadline (const size_t Limit = DefaultHistoryLimit)
00432       : HistoryLimit (Limit), HistoryFileName (""),
00433         OriginalCompletion (rl_attempted_completion_function) {
00434       rl_startup_hook = StartupHook;
00435       rl_attempted_completion_function = UserCompletion;
00436       using_history();
00437     }
00438
00446     SReadline (const std::string&  historyFileName,
00447                const size_t Limit = DefaultHistoryLimit)
00448       : HistoryLimit (Limit), HistoryFileName (historyFileName),
00449         OriginalCompletion (rl_attempted_completion_function) {
00450       rl_startup_hook = StartupHook;
00451       rl_attempted_completion_function = UserCompletion;
00452       using_history();
00453       LoadHistory (HistoryFileName);
00454     }
00455
00460     ~SReadline() {
00461       rl_attempted_completion_function = OriginalCompletion;
00462       SaveHistory (HistoryFileName);
00463     }
00464
00471     std::string GetLine (const std::string& Prompt) {
00472       bool Unused;
00473       return GetLine (Prompt, Unused);
00474     }
00475
00484     template <typename  Container>
00485     std::string GetLine (const std::string& Prompt, Container& ReadTokens) {
00486       bool Unused;
00487       return GetLine (Prompt, ReadTokens, Unused);
00488     }
00489
00499     template <typename  Container>
00500     std::string GetLine (const std::string& Prompt, Container& ReadTokens,
00501                          bool& BreakOut) {
00502       std::string Input (GetLine (Prompt, BreakOut));
00503       SplitTokens (Input, ReadTokens);
00504       return Input;
00505     }
00506
00507
00515     std::string GetLine (const std::string& Prompt, bool& BreakOut) {
00516       BreakOut = true;
00517
00518       char* ReadLine (readline (Prompt.c_str()));
00519       if (ReadLine == NULL) {
00520         return std::string();
00521       }
00522
00523       // It's OK
```

```
00524       BreakOut = false;
00525       std::string Input (ReadLine);
00526       free (ReadLine); ReadLine = NULL;
00527
00528       boost::algorithm::trim (Input);
00529       if (Input.empty() == false) {
00530         if (history_length == 0
00531             || Input != history_list()[ history_length - 1 ]->line) {
00532           add_history (Input.c_str());
00533
00534           if (history_length >= static_cast<int> (HistoryLimit)) {
00535             stifle_history (HistoryLimit);
00536           }
00537         }
00538       }
00539
00540       return Input;
00541     }
00542
00543
00549     template <typename ContainerType>
00550     void GetHistory (ContainerType&  Container) {
00551       for (int k (0); k < history_length; ++k ) {
00552         Container.push_back (history_list()[k]->line);
00553       }
00554     }
00555
00562     bool SaveHistory (std::ostream& OS) {
00563       if (!OS) {
00564         return false;
00565       }
00566
00567       for (int k (0); k < history_length; ++k) {
00568         OS << history_list()[ k ]->line << std::endl;
00569       }
00570       return true;
00571     }
00572
00579     bool SaveHistory (const std::string&  FileName) {
00580       if (FileName.empty() == true) {
00581         return false;
00582       }
00583
00584       std::ofstream OS (FileName.c_str());
00585       return SaveHistory (OS);
00586     }
00587
00592     void ClearHistory() {
00593       clear_history();
00594     }
00595
00602     bool LoadHistory (std::istream& IS) {
00603       if (!IS) {
00604         return false;
00605       }
00606
00607       ClearHistory();
00608       std::string OneLine;
00609
00610       while (!getline (IS, OneLine).eof()) {
00611         boost::algorithm::trim( OneLine );
00612         if ((history_length == 0)
00613             || OneLine != history_list()[history_length - 1]->line) {
00614           add_history (OneLine.c_str());
00615         }
00616       }
00617       stifle_history (HistoryLimit);
00618       return true;
00619     }
00620
00627     bool LoadHistory (const std::string&  FileName) {
00628       if (FileName.empty() == true) {
00629         return false;
00630       }
```

```
00631
00632       std::ifstream IS (FileName.c_str());
00633       return LoadHistory (IS);
00634     }
00635
00655     template <typename  ContainerType>
00656     void RegisterCompletions (const ContainerType& Container) {
00657       Completions.clear();
00658       for (typename ContainerType::const_iterator k (Container.begin());
00659            k != Container.end(); ++k) {
00660         std::vector<std::string> OneLine;
00661         const std::string& kStr = static_cast<std::string> (*k);
00662
00663         SplitTokens (kStr, OneLine);
00664         Completions.push_back (OneLine);
00665       }
00666     }
00667
00673     void SetKeymap (SKeymap& NewKeymap) {
00674       rl_set_keymap (NewKeymap.keymap);
00675       KeymapWasSetup = true;
00676       Earlykeymap = NewKeymap.keymap;
00677     }
00678
00679
00680   private:
00681     // ////////////////////////// Attributes ////////////////////////
00685     const size_t HistoryLimit;
00686
00690     const std::string HistoryFileName;
00691
00695     rl_completion_func_t* OriginalCompletion;
00696   };
00697
00698 }; // namespace swift
00699
00700 #endif
00701
```

25.57 test/sevmgr/EventQueueManagementTestSuite.cpp File Reference

25.58 EventQueueManagementTestSuite.cpp

```
00001
00005 // ////////////////////////////////////////////////////////////////////
00006 // Import section
00007 // ////////////////////////////////////////////////////////////////////
00008 // STL
00009 #include <sstream>
00010 #include <fstream>
00011 #include <map>
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 EventQueueManagementTest
00017 #include <boost/test/unit_test.hpp>
00018 // StdAir
00019 #include <stdair/stdair_basic_types.hpp>
00020 #include <stdair/basic/BasLogParams.hpp>
00021 #include <stdair/basic/BasDBParams.hpp>
00022 #include <stdair/basic/BasFileMgr.hpp>
00023 #include <stdair/basic/ProgressStatusSet.hpp>
00024 #include <stdair/bom/EventStruct.hpp>
00025 #include <stdair/bom/EventQueue.hpp>
00026 #include <stdair/bom/BookingRequestStruct.hpp>
00027 #include <stdair/service/Logger.hpp>
00028 // SEvMgr
00029 #include <sevmgr/SEVMGR_Service.hpp>
00030 #include <sevmgr/config/sevmgr-paths.hpp>
```

```
00031
00032 namespace boost_utf = boost::unit_test;
00033
00034 // (Boost) Unit Test XML Report
00035 std::ofstream utfReportStream ("EventQueueManagementTestSuite_utfresults.xml");
00036
00040 struct UnitTestConfig {
00042   UnitTestConfig() {
00043     boost_utf::unit_test_log.set_stream (utfReportStream);
00044     boost_utf::unit_test_log.set_format (boost_utf::XML);
00045     boost_utf::unit_test_log.set_threshold_level (boost_utf::log_test_units);
00046     //boost_utf::unit_test_log.set_threshold_level
       (boost_utf::log_successful_tests);
00047   }
00048
00050   ~UnitTestConfig() {
00051   }
00052 };
00053
00054 // Specific type definitions
00055 typedef std::pair<stdair::Count_T, stdair::Count_T> NbOfEventsPair_T;
00056 typedef std::map<const stdair::DemandStreamKeyStr_T,
00057                  NbOfEventsPair_T> NbOfEventsByDemandStreamMap_T;
00058
00059
00060 // /////////////// Main: Unit Test Suite //////////////
00061
00062 // Set the UTF configuration (re-direct the output to a specific file)
00063 BOOST_GLOBAL_FIXTURE (UnitTestConfig);
00064
00065 // Start the test suite
00066 BOOST_AUTO_TEST_SUITE (master_test_suite)
00067
00068
00071 BOOST_AUTO_TEST_CASE (sevmgr_simple_simulation_test) {
00072
00073   // Input file name
00074   const stdair::Filename_T lInputFilename (STDAIR_SAMPLE_DIR "/demand01.csv");
00075
00076   // Check that the file path given as input corresponds to an actual file
00077   const bool doesExistAndIsReadable =
00078     stdair::BasFileMgr::doesExistAndIsReadable (lInputFilename);
00079   BOOST_CHECK_MESSAGE (doesExistAndIsReadable == true,
00080                        "The '" << lInputFilename
00081                        << "' input file can not be open and read");
00082
00083   // Output log File
00084   const stdair::Filename_T lLogFilename ("EventQueueManagementTestSuite.log");
00085
00086   // Set the log parameters
00087   std::ofstream logOutputFile;
00088   // open and clean the log outputfile
00089   logOutputFile.open (lLogFilename.c_str());
00090   logOutputFile.clear();
00091
00092   // Initialise the Sevmgr service object
00093   const stdair::BasLogParams lLogParams (stdair::LOG::DEBUG, logOutputFile);
00094   SEVMGR::SEVMGR_Service sevmgrService (lLogParams);
00095
00097   sevmgrService.buildSampleBom();
00098
00101   sevmgrService.reset();
00102
00103   // DEBUG
00104   STDAIR_LOG_DEBUG ("End of the simulation");
00105
00106   // Close the log file
00107   logOutputFile.close();
00108 }
00109
00110 // End the test suite
00111 BOOST_AUTO_TEST_SUITE_END()
00112
00113
```

Index

~FacSEVMGRServiceContext
SEVMGR::FacSEVMGRService-Context, 132
~PYEventQueueManager
SEVMGR::PYEventQueueManager, 144
~SEVMGR_Service
SEVMGR::SEVMGR_Service, 151
~SKeymap
swift::SKeymap, 155
~SReadline
swift::SReadline, 158
BINDIR
sevmgr-paths.hpp, 178
sevmgr-paths.hpp.in, 181
Bind
swift::SKeymap, 155
COMMAND, 123
doc, 124
func, 123
name, 123
ClearHistory
swift::SReadline, 160
CmdAbstract, 123
DATADIR
sevmgr-paths.hpp, 179
sevmgr-paths.hpp.in, 181
DATAROOTDIR
sevmgr-paths.hpp, 179
sevmgr-paths.hpp.in, 181
DOCDIR
sevmgr-paths.hpp, 179
sevmgr-paths.hpp.in, 181
EXEC_PREFIX
sevmgr-paths.hpp, 178
sevmgr-paths.hpp.in, 181
FacSEVMGRServiceContext
SEVMGR::FacSEVMGRService-Context, 132
SEVMGR::SEVMGR_Service-Context, 153
FacServiceAbstract, 131
GetHistory
swift::SReadline, 159
GetLine
swift::SReadline, 158, 159
HTMLDIR
sevmgr-paths.hpp, 179
sevmgr-paths.hpp.in, 181
INCLUDEDIR
sevmgr-paths.hpp, 178
sevmgr-paths.hpp.in, 181
INFODIR
sevmgr-paths.hpp, 179
sevmgr-paths.hpp.in, 181
LIBDIR
sevmgr-paths.hpp, 178
sevmgr-paths.hpp.in, 181
LIBEXECDIR
sevmgr-paths.hpp, 178
sevmgr-paths.hpp.in, 181
LoadHistory
swift::SReadline, 160
MANDIR
sevmgr-paths.hpp, 179
sevmgr-paths.hpp.in, 181
PACKAGE
sevmgr-paths.hpp, 178
sevmgr-paths.hpp.in, 180
PACKAGE_NAME
sevmgr-paths.hpp, 178
sevmgr-paths.hpp.in, 180
PACKAGE_VERSION
sevmgr-paths.hpp, 178
sevmgr-paths.hpp.in, 180
PDFDIR
sevmgr-paths.hpp, 179
sevmgr-paths.hpp.in, 182
PREFIXDIR
sevmgr-paths.hpp, 178
sevmgr-paths.hpp.in, 180
PYEventQueueManager
SEVMGR::PYEventQueueManager, 144
RegisterCompletions
swift::SReadline, 161
RootException, 147
SBINDIR
sevmgr-paths.hpp, 178
sevmgr-paths.hpp.in, 181
SEVMGR, 113
SEVMGR_ServicePtr_T, 115

bounded1_2_p_t, 115
bounded1_3_p_t, 115
bounded1_4_p_t, 115
bounded2_p_t, 115
bounded4_p_t, 115
char_t, 114
chset_t, 115
int1_p_t, 114
iterator_t, 114
repeat_p_t, 115
rule_t, 114
scanner_t, 114
uint1_2_p_t, 114
uint1_3_p_t, 114
uint1_4_p_t, 115
uint2_p_t, 114
uint4_p_t, 114
SEVMGR::EventQueueManager, 129
SEVMGR_Service, 130
SEVMGR::FacSEVMGRServiceContext, 131
create, 132
instance, 132
SEVMGR::PYEventQueueManager, 143
~PYEventQueueManager, 144
PYEventQueueManager, 144
init, 144
sevmgr, 144
SEVMGR::SEVMGR_Service, 149
buildSampleBom, 151
csvDisplay, 152
isQueueDone, 151
popEvent, 151
reset, 151
SEVMGR::SEVMGR_ServiceContext, 152
SEVMGR::SEvMgrException, 153
SEvMgrException, 153
SEVMGR_Service
SEVMGR::EventQueueManager, 130
SEVMGR::SEVMGR_Service, 150
SEVMGR::SEVMGR_Service-Context, 152
SEVMGR_ServicePtr_T
SEVMGR, 115
SEvMgrException
SEVMGR::SEvMgrException, 153
SKeymap
swift::SKeymap, 154, 155
SReadline
swift::SKeymap, 156
swift::SReadline, 157
STDAIR_SAMPLE_DIR
sevmgr-paths.hpp, 179
sevmgr-paths.hpp.in, 182
SYSCONFDIR
sevmgr-paths.hpp, 178
sevmgr-paths.hpp.in, 181
SaveHistory
swift::SReadline, 159, 160
ServiceAbstract, 148
SetKeymap
swift::SReadline, 161
Unbind
swift::SKeymap, 155

bounded1_2_p_t
SEVMGR, 115
bounded1_3_p_t
SEVMGR, 115
bounded1_4_p_t
SEVMGR, 115
bounded2_p_t
SEVMGR, 115
bounded4_p_t
SEVMGR, 115
buildSampleBom
SEVMGR::SEVMGR_Service, 151

char_t
SEVMGR, 114
chset_t
SEVMGR, 115
com_cd
readline_autocomp.hpp, 199
com_delete
readline_autocomp.hpp, 198
com_help
readline_autocomp.hpp, 198
com_list
readline_autocomp.hpp, 198
com_pwd
readline_autocomp.hpp, 198
com_quit
readline_autocomp.hpp, 199
com_rename
readline_autocomp.hpp, 198
com_stat
readline_autocomp.hpp, 198

com_view
readline_autocomp.hpp, 198
command_generator
readline_autocomp.hpp, 199
commands
readline_autocomp.hpp, 200
create
SEVMGR::FacSEVMGRService-Context, 132
csvDisplay
SEVMGR::SEVMGR_Service, 152

doc
COMMAND, 124
doc/local/authors.doc, 168
doc/local/codingrules.doc, 168
doc/local/copyright.doc, 168
doc/local/documentation.doc, 168
doc/local/features.doc, 168
doc/local/help_wanted.doc, 168
doc/local/howto_release.doc, 168
doc/local/index.doc, 168
doc/local/installation.doc, 168
doc/local/linking.doc, 169
doc/local/test.doc, 169
doc/local/users_guide.doc, 169
doc/local/verification.doc, 169
doc/tutorial/tutorial.doc, 169
done
readline_autocomp.hpp, 200
dupstr
readline_autocomp.hpp, 199

execute_line
readline_autocomp.hpp, 199

fileman_completion
readline_autocomp.hpp, 199
find_command
readline_autocomp.hpp, 199
func
COMMAND, 123

getwd
readline_autocomp.hpp, 198

init
SEVMGR::PYEventQueueManager, 144
initialize_readline
readline_autocomp.hpp, 200
instance
SEVMGR::FacSEVMGRService-Context, 132
int1_p_t
SEVMGR, 114
isQueueDone
SEVMGR::SEVMGR_Service, 151
iterator_t
SEVMGR, 114

main
sevmgr_demo.cpp, 173

name
COMMAND, 123

operator=
swift::SKeymap, 155

popEvent
SEVMGR::SEVMGR_Service, 151
pt2Func
readline_autocomp.hpp, 197

readConfiguration
sevmgr_demo.cpp, 173
readline_autocomp.hpp
com_cd, 199
com_delete, 198
com_help, 198
com_list, 198
com_pwd, 198
com_quit, 199
com_rename, 198
com_stat, 198
com_view, 198
command_generator, 199
commands, 200
done, 200
dupstr, 199
execute_line, 199
fileman_completion, 199
find_command, 199
getwd, 198
initialize_readline, 200
pt2Func, 197
stripwhite, 199
syscom, 201
too_dangerous, 200
valid_argument, 200
xmalloc, 198

repeat_p_t
SEVMGR, 115
reset
SEVMGR::SEVMGR_Service, 151
rule_t
SEVMGR, 114

scanner_t
SEVMGR, 114
sevmgr
SEVMGR::PYEventQueueManager, 144
sevmgr-paths.hpp
BINDIR, 178
DATADIR, 179
DATAROOTDIR, 179
DOCDIR, 179
EXEC_PREFIX, 178
HTMLDIR, 179
INCLUDEDIR, 178
INFODIR, 179
LIBDIR, 178
LIBEXECDIR, 178
MANDIR, 179
PACKAGE, 178
PACKAGE_NAME, 178
PACKAGE_VERSION, 178
PDFDIR, 179
PREFIXDIR, 178
SBINDIR, 178
STDAIR_SAMPLE_DIR, 179
SYSCONFDIR, 178
sevmgr-paths.hpp.in
BINDIR, 181
DATADIR, 181
DATAROOTDIR, 181
DOCDIR, 181
EXEC_PREFIX, 181
HTMLDIR, 181
INCLUDEDIR, 181
INFODIR, 181
LIBDIR, 181
LIBEXECDIR, 181
MANDIR, 181
PACKAGE, 180
PACKAGE_NAME, 180
PACKAGE_VERSION, 180
PDFDIR, 182
PREFIXDIR, 180
SBINDIR, 181
STDAIR_SAMPLE_DIR, 182
SYSCONFDIR, 181
sevmgr/ Directory Reference, 112
sevmgr/SEVMGR_Exceptions.hpp, 193, 194
sevmgr/SEVMGR_Service.hpp, 194, 195
sevmgr/SEVMGR_Types.hpp, 196
sevmgr/basic/ Directory Reference, 110
sevmgr/basic/BasConst.cpp, 169
sevmgr/basic/BasConst_EventQueue-Manager.hpp, 169
sevmgr/basic/BasConst_SEVMGR_-Service.hpp, 170
sevmgr/basic/BasParserTypes.hpp, 170, 171
sevmgr/batches/ Directory Reference, 111
sevmgr/batches/sevmgr_demo.cpp, 172, 174
sevmgr/command/ Directory Reference, 111
sevmgr/command/EventQueueManager.-cpp, 176
sevmgr/command/EventQueueManager.-hpp, 176, 177
sevmgr/config/ Directory Reference, 111
sevmgr/config/sevmgr-paths.hpp, 177, 179
sevmgr/config/sevmgr-paths.hpp.in, 180, 182
sevmgr/factory/ Directory Reference, 111
sevmgr/factory/FacSEVMGRService-Context.cpp, 182
sevmgr/factory/FacSEVMGRService-Context.hpp, 183
sevmgr/python/ Directory Reference, 112
sevmgr/python/pysevmgr.cpp, 184, 185
sevmgr/service/ Directory Reference, 112
sevmgr/service/SEVMGR_Service.cpp, 187
sevmgr/service/SEVMGR_Service-Context.cpp, 191
sevmgr/service/SEVMGR_Service-Context.hpp, 192
sevmgr/ui/ Directory Reference, 113
sevmgr/ui/cmdline/ Directory Reference, 111
sevmgr/ui/cmdline/SReadline.hpp, 216, 217
sevmgr/ui/cmdline/readline_autocomp.-hpp, 196, 201

sevmgr/ui/cmdline/sevmgr.cpp, 205
sevmgr_demo.cpp
    main, 173
    readConfiguration, 173
std::allocator, 116
std::auto_ptr, 116
std::bad_alloc, 116
std::bad_cast, 117
std::bad_exception, 117
std::bad_typeid, 118
std::basic_fstream, 118
std::basic_ifstream, 118
std::basic_ios, 119
std::basic_iostream, 119
std::basic_istream, 120
std::basic_istringstream, 120
std::basic_ofstream, 120
std::basic_ostream, 121
std::basic_ostringstream, 121
std::basic_string, 122
std::basic_string::const_iterator, 125
std::basic_string::const_reverse_iterator, 127
std::basic_string::iterator, 137
std::basic_string::reverse_iterator, 146
std::basic_stringstream, 122
std::bitset, 122
std::complex, 124
std::deque, 129
std::deque::const_iterator, 126
std::deque::const_reverse_iterator, 128
std::deque::iterator, 136
std::deque::reverse_iterator, 146
std::domain_error, 129
std::exception, 130
std::fstream, 133
std::ifstream, 133
std::invalid_argument, 134
std::ios, 134
std::ios_base, 135
std::ios_base::failure, 133
std::istream, 135
std::istringstream, 135
std::length_error, 138
std::list, 139
std::list::const_iterator, 124
std::list::const_reverse_iterator, 127
std::list::iterator, 136
std::list::reverse_iterator, 145
std::logic_error, 139
std::map, 139
std::map::const_iterator, 124
std::map::const_reverse_iterator, 126
std::map::iterator, 138
std::map::reverse_iterator, 146
std::multimap, 140
std::multimap::const_iterator, 125
std::multimap::const_reverse_iterator, 127
std::multimap::iterator, 136
std::multimap::reverse_iterator, 146
std::multiset, 140
std::multiset::const_iterator, 125
std::multiset::const_reverse_iterator, 127
std::multiset::iterator, 137
std::multiset::reverse_iterator, 147
std::ofstream, 141
std::ostream, 141
std::ostringstream, 142
std::out_of_range, 142
std::overflow_error, 143
std::priority_queue, 143
std::queue, 144
std::range_error, 145
std::runtime_error, 148
std::set, 148
std::set::const_iterator, 125
std::set::const_reverse_iterator, 127
std::set::iterator, 137
std::set::reverse_iterator, 147
std::stack, 161
std::string, 162
std::string::const_iterator, 126
std::string::const_reverse_iterator, 128
std::string::iterator, 137
std::string::reverse_iterator, 147
std::stringstream, 162
std::underflow_error, 163
std::valarray, 163
std::vector, 163
std::vector::const_iterator, 125
std::vector::const_reverse_iterator, 128
std::vector::iterator, 137
std::vector::reverse_iterator, 145
std::wfstream, 164
std::wifstream, 164
std::wios, 165
std::wistream, 165
std::wistringstream, 165
std::wofstream, 166
std::wostream, 166

std::wostringstream, 167
std::wstring, 167
std::wstring::const_iterator, 126
std::wstring::const_reverse_iterator, 128
std::wstring::iterator, 138
std::wstring::reverse_iterator, 146
std::wstringstream, 168
stdair, 116
stripwhite
    readline_autocomp.hpp, 199
swift, 116
swift::SKeymap, 154
    ~SKeymap, 155
    Bind, 155
    SKeymap, 154, 155
    SReadline, 156
    Unbind, 155
    operator=, 155
swift::SReadline, 156
    ~SReadline, 158
    ClearHistory, 160
    GetHistory, 159
    GetLine, 158, 159
    LoadHistory, 160
    RegisterCompletions, 161
    SReadline, 157
    SaveHistory, 159, 160
    SetKeymap, 161
syscom
    readline_autocomp.hpp, 201

test/ Directory Reference, 112
test/sevmgr/ Directory Reference, 112
test/sevmgr/EventQueueManagement-
        TestSuite.cpp, 223
too_dangerous
    readline_autocomp.hpp, 200

uint1_2_p_t
    SEVMGR, 114
uint1_3_p_t
    SEVMGR, 114
uint1_4_p_t
    SEVMGR, 115
uint2_p_t
    SEVMGR, 114
uint4_p_t
    SEVMGR, 114

valid_argument
    readline_autocomp.hpp, 200

xmalloc
    readline_autocomp.hpp, 198