

SimCRS Reference Manual

1.00.0

Generated by Doxygen 1.4.7

Wed Dec 26 09:31:07 2012

Contents

1	SimCRS Documentation	1
2	SimCRS Directory Hierarchy	2
3	SimCRS Namespace Index	3
4	SimCRS Hierarchical Index	3
5	SimCRS Class Index	5
6	SimCRS File Index	6
7	SimCRS Page Index	7
8	SimCRS Directory Documentation	7
9	SimCRS Namespace Documentation	9
10	SimCRS Class Documentation	11
11	SimCRS File Documentation	34
12	SimCRS Page Documentation	48

1 SimCRS Documentation

1.1 Getting Started

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

1.2 SimCRS at SourceForge

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

1.3 SimCRS Development

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

1.4 External Libraries

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

1.5 Support SimCRS

1.6 About SimCRS

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

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

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

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

Note:

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

2 SimCRS Directory Hierarchy

2.1 SimCRS Directories

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

simcrs	9
basic	7
batches	8
bom	8
command	8
config	8
factory	8
service	8
test	9
simcrs	9

3 SimCRS Namespace Index

3.1 SimCRS Namespace List

Here is a list of all namespaces with brief descriptions:

AIRINV	9
SIMCRS	9
stdair (Forward declarations)	10

4 SimCRS Hierarchical Index

4.1 SimCRS Class Hierarchy

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

std::allocator< T >

```

std::auto_ptr< T >
std::basic_string< Char >
std::basic_string< char >
    std::string
std::basic_string< wchar_t >
    std::wstring
std::bitset< Bits >

```

SIMCRS::BomAbstract**11**

```

std::complex
std::deque< T >

```

SIMCRS::DistributionManager**13**

```

std::exception
    std::bad_alloc
    std::bad_cast
    std::bad_exception
    std::bad_typeid
    std::ios_base::failure
    std::logic_error
        std::domain_error
        std::invalid_argument
        std::length_error
        std::out_of_range
    std::runtime_error
        std::overflow_error
        std::range_error
        std::underflow_error

```

SIMCRS::FacBomAbstract**14****SIMCRS::FacServiceAbstract****16****SIMCRS::FacSimcrsServiceContext****18****SIMCRS::FacSupervisor****20**

```

std::ios_base
    std::basic_ios
        std::basic_istream
            std::basic_ifstream
            std::basic_iostream
                std::basic_fstream
                std::basic_stringstream
            std::basic_istreamstream
        std::basic_ostream
            std::basic_iostream
            std::basic_ofstream
            std::basic_ostringstream
    std::basic_ios< char >
        std::basic_istream< char >
            std::basic_ifstream< char >
            std::ifstream
        std::basic_iostream< char >
            std::basic_fstream< char >
            std::fstream

```

std::basic_stringstream< char >	
std::stringstream	
std::basic_istream< char >	
std::istream	
std::istream	
std::basic_ostream< char >	
std::basic_iostream< char >	
std::basic_ofstream< char >	
std::ofstream	
std::basic_ostringstream< char >	
std::ostringstream	
std::ostream	
std::ios	
std::basic_ios< wchar_t >	
std::basic_istream< wchar_t >	
std::basic_ifstream< wchar_t >	
std::wifstream	
std::basic_iostream< wchar_t >	
std::basic_fstream< wchar_t >	
std::wfstream	
std::basic_stringstream< wchar_t >	
std::wstringstream	
std::basic_istream< wchar_t >	
std::wistream	
std::wistream	
std::basic_ostream< wchar_t >	
std::basic_iostream< wchar_t >	
std::basic_ofstream< wchar_t >	
std::wofstream	
std::basic_ostringstream< wchar_t >	
std::wostringstream	
std::wostream	
std::wios	
std::list< T >	
std::map< K, T >	
std::multimap< K, T >	
std::multiset< K >	
std::priority_queue< T >	
std::queue< T >	
RootException	23
SIMCRS::AvailabilityRetrievalException	11
SIMCRS::BookingException	13
SIMCRS::ServiceAbstract	23
SIMCRS::SIMCRS_ServiceContext	32
std::set< K >	
SIMCRS::SIMCRS_Service	24
std::stack< T >	
std::valarray< T >	

`std::vector< T >`

5 SimCRS Class Index

5.1 SimCRS Class List

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

SIMCRS::AvailabilityRetrievalException	11
SIMCRS::BomAbstract	11
SIMCRS::BookingException	13
SIMCRS::DistributionManager (Command wrapping the travel distribution (CRS/GDS) process)	13
SIMCRS::FacBomAbstract	14
SIMCRS::FacServiceAbstract	16
SIMCRS::FacSimcrsServiceContext	18
SIMCRS::FacSupervisor	20
RootException	23
SIMCRS::ServiceAbstract	23
SIMCRS::SIMCRS_Service	24
SIMCRS::SIMCRS_ServiceContext (Class holding the context of the Simcrs services)	32

6 SimCRS File Index

6.1 SimCRS File List

Here is a list of all files with brief descriptions:

simcrs/SIMCRS_Service.hpp	47
simcrs/SIMCRS_Types.hpp	48
simcrs/basic/BasConst.cpp	34
simcrs/basic/BasConst_General.hpp	35
simcrs/basic/BasConst_SIMCRS_Service.hpp	35
simcrs/batches/simcrs.cpp	35
simcrs/bom/BomAbstract.cpp	38

simcrs/bom/BomAbstract.hpp	39
simcrs/command/DistributionManager.cpp	39
simcrs/command/DistributionManager.hpp	40
simcrs/config/simcrs-paths.hpp.in	40
simcrs/factory/FacBomAbstract.cpp	42
simcrs/factory/FacBomAbstract.hpp	43
simcrs/factory/FacServiceAbstract.cpp	43
simcrs/factory/FacServiceAbstract.hpp	43
simcrs/factory/FacSimcrsServiceContext.cpp	43
simcrs/factory/FacSimcrsServiceContext.hpp	44
simcrs/factory/FacSupervisor.cpp	44
simcrs/factory/FacSupervisor.hpp	44
simcrs/service/ServiceAbstract.cpp	44
simcrs/service/ServiceAbstract.hpp	45
simcrs/service/SIMCRS_Service.cpp	45
simcrs/service/SIMCRS_ServiceContext.cpp	46
simcrs/service/SIMCRS_ServiceContext.hpp	46
test/simcrs/CRSTestSuite.cpp	48

7 SimCRS Page Index

7.1 SimCRS Related Pages

Here is a list of all related documentation pages:

People	48
Coding Rules	49
Copyright and License	50
Documentation Rules	56
Main features	58
Make a Difference	59
Make a new release	59

Installation	62
Linking with SimCRS	73
Test Rules	75
Users Guide	76
Supported Systems	127
SimCRS Supported Systems (Previous Releases)	128
Tutorials	132
Command-Line Test to Demonstrate How To Test the SimCRS Project	135

8 SimCRS Directory Documentation

8.1 simcrs/basic/ Directory Reference

Files

- file [BasConst.cpp](#)
- file [BasConst_General.hpp](#)
- file [BasConst_SIMCRS_Service.hpp](#)

8.2 simcrs/batches/ Directory Reference

Files

- file [simcrs.cpp](#)

8.3 simcrs/bom/ Directory Reference

Files

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

8.4 simcrs/command/ Directory Reference

Files

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

8.5 simcrs/config/ Directory Reference

Files

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

8.6 simcrs/factory/ Directory Reference

Files

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

8.7 simcrs/service/ Directory Reference

Files

- file [ServiceAbstract.cpp](#)
- file [ServiceAbstract.hpp](#)
- file [SIMCRS_Service.cpp](#)
- file [SIMCRS_ServiceContext.cpp](#)
- file [SIMCRS_ServiceContext.hpp](#)

8.8 test/simcrs/ Directory Reference

Files

- file [CRSTestSuite.cpp](#)

8.9 simcrs/ Directory Reference

Directories

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

Files

- file [SIMCRS_Service.hpp](#)
- file [SIMCRS_Types.hpp](#)

8.10 test/ Directory Reference

Directories

- directory [simcrs](#)

9 SimCRS Namespace Documentation

9.1 AIRINV Namespace Reference

9.2 SIMCRS Namespace Reference

Classes

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

Typedefs

- typedef std::string [CRSCode_T](#)
- typedef boost::shared_ptr< [SIMCRS_Service](#) > [SIMCRS_ServicePtr_T](#)

Variables

- const std::string [DEFAULT_CRIS_CODE](#) = "IS"
- const std::string [DEFAULT_CRIS_CODE](#)

9.2.1 Typedef Documentation

9.2.1.1 typedef std::string [SIMCRS::CRSCode_T](#)

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

Definition at line 39 of file SIMCRS_Types.hpp.

9.2.1.2 typedef boost::shared_ptr<[SIMCRS_Service](#)> [SIMCRS::SIMCRS_ServicePtr_T](#)

(Smart) Pointer on the SimCRS service handler.

Definition at line 44 of file SIMCRS_Types.hpp.

9.2.2 Variable Documentation

9.2.2.1 const std::string [SIMCRS::DEFAULT_CRS_CODE](#) = "1S"

Default CRS code for the [SIMCRS_Service](#).

Definition at line 10 of file BasConst.cpp.

9.2.2.2 const std::string [SIMCRS::DEFAULT_CRS_CODE](#)

Default CRS code for the [SIMCRS_Service](#).

Definition at line 10 of file BasConst.cpp.

9.3 stdair Namespace Reference

Forward declarations.

9.3.1 Detailed Description

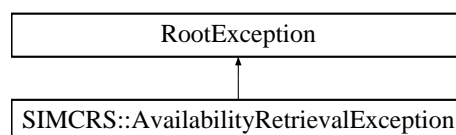
Forward declarations.

10 SimCRS Class Documentation

10.1 SIMCRS::AvailabilityRetrievalException Class Reference

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

Inheritance diagram for SIMCRS::AvailabilityRetrievalException::



10.1.1 Detailed Description

Specific exception related to availability calculation.

Definition at line 31 of file SIMCRS_Types.hpp.

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

- [simcrs/SIMCRS_Types.hpp](#)

10.2 SIMCRS::BomAbstract Class Reference

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

Public Member Functions

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

Protected Member Functions

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

Friends

- class [FacBomAbstract](#)

10.2.1 Detailed Description

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

Definition at line 14 of file BomAbstract.hpp.

10.2.2 Constructor & Destructor Documentation

10.2.2.1 SIMCRS::BomAbstract::BomAbstract () [inline, protected]

Protected Default Constructor to ensure this class is abstract.

Definition at line 40 of file BomAbstract.hpp.

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

Definition at line 41 of file BomAbstract.hpp.

10.2.2.3 `virtual SIMCRS::BomAbstract::~~BomAbstract () [inline, protected, virtual]`

Destructor.

Definition at line 44 of file BomAbstract.hpp.

10.2.3 Member Function Documentation

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

Dump a Business Object into an output stream.

Parameters:

ostream& the output stream.

10.2.3.2 `virtual void SIMCRS::BomAbstract::fromStream (std::istream & ioIn) [pure virtual]`

Read a Business Object from an input stream.

Parameters:

istream& the input stream.

Referenced by operator>>().

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

Get the serialised version of the Business Object.

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

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

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

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

10.2.4 Friends And Related Function Documentation

10.2.4.1 `friend class FacBomAbstract [friend]`

Definition at line 15 of file BomAbstract.hpp.

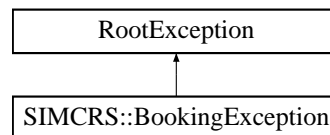
The documentation for this class was generated from the following file:

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

10.3 SIMCRS::BookingException Class Reference

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

Inheritance diagram for SIMCRS::BookingException::



10.3.1 Detailed Description

Specific exception related to bookings made against the CRS.

Definition at line 25 of file SIMCRS_Types.hpp.

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

- [simcrs/SIMCRS_Types.hpp](#)

10.4 SIMCRS::DistributionManager Class Reference

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

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

Friends

- class [SIMCRS_Service](#)

10.4.1 Detailed Description

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

Definition at line 30 of file DistributionManager.hpp.

10.4.2 Friends And Related Function Documentation

10.4.2.1 friend class [SIMCRS_Service](#) [friend]

Definition at line 31 of file DistributionManager.hpp.

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

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

10.5 SIMCRS::FacBomAbstract Class Reference

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

Public Types

- typedef std::vector< [BomAbstract](#) * > [BomPool_T](#)

Static Public Member Functions

- static std::size_t [getID](#) (const [BomAbstract](#) *)
- static std::size_t [getID](#) (const [BomAbstract](#) &)
- static std::string [getIDString](#) (const [BomAbstract](#) *)
- static std::string [getIDString](#) (const [BomAbstract](#) &)

Protected Member Functions

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

Protected Attributes

- [BomPool_T _pool](#)

Friends

- class [FacSupervisor](#)

10.5.1 Detailed Description

Base class for Factory layer.

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

10.5.2 Member Typedef Documentation

10.5.2.1 typedef std::vector<[BomAbstract](#)*> [SIMCRS::FacBomAbstract::BomPool_T](#)

Define the list (pool) of Bom objects.

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

10.5.3 Constructor & Destructor Documentation

10.5.3.1 [SIMCRS::FacBomAbstract::FacBomAbstract](#) () [inline, protected]

Default Constructor.

This constructor is protected to ensure the class is abstract.

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

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

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

10.5.3.3 SIMCRS::FacBomAbstract::~~FacBomAbstract () [protected, virtual]

Destructor.

Definition at line 16 of file FacBomAbstract.cpp.

10.5.4 Member Function Documentation

10.5.4.1 std::size_t SIMCRS::FacBomAbstract::getID (const BomAbstract *) [static]

Return the ID corresponding to the given object pointer.

Definition at line 35 of file FacBomAbstract.cpp.

Referenced by getID(), and getIDString().

10.5.4.2 std::size_t SIMCRS::FacBomAbstract::getID (const BomAbstract &) [static]

Return the ID corresponding to the given object reference.

Definition at line 43 of file FacBomAbstract.cpp.

References getID().

10.5.4.3 std::string SIMCRS::FacBomAbstract::getIDString (const BomAbstract *) [static]

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

Definition at line 48 of file FacBomAbstract.cpp.

References getID().

Referenced by getIDString().

10.5.4.4 std::string SIMCRS::FacBomAbstract::getIDString (const BomAbstract &) [static]

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

Definition at line 56 of file FacBomAbstract.cpp.

References getIDString().

10.5.5 Friends And Related Function Documentation

10.5.5.1 friend class FacSupervisor [friend]

Definition at line 18 of file FacBomAbstract.hpp.

10.5.6 Member Data Documentation

10.5.6.1 BomPool_T SIMCRS::FacBomAbstract::_pool [protected]

List of instantiated Business Objects

Definition at line 53 of file FacBomAbstract.hpp.

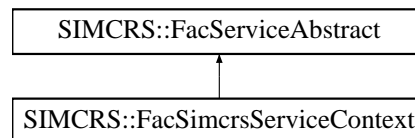
The documentation for this class was generated from the following files:

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

10.6 SIMCRS::FacServiceAbstract Class Reference

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

Inheritance diagram for SIMCRS::FacServiceAbstract::



Public Types

- `typedef std::vector< ServiceAbstract * > ServicePool_T`

Public Member Functions

- virtual `~FacServiceAbstract ()`
- void `clean ()`

Protected Member Functions

- `FacServiceAbstract ()`

Protected Attributes

- `ServicePool_T _pool`

10.6.1 Detailed Description

Base class for the (Service) Factory layer.

Definition at line 16 of file `FacServiceAbstract.hpp`.

10.6.2 Member Typedef Documentation

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

Define the list (pool) of Service objects.

Definition at line 20 of file `FacServiceAbstract.hpp`.

10.6.3 Constructor & Destructor Documentation

10.6.3.1 SIMCRS::FacServiceAbstract::~~FacServiceAbstract () [virtual]

Destructor.

Definition at line 13 of file FacServiceAbstract.cpp.

References `clean()`.

10.6.3.2 SIMCRS::FacServiceAbstract::FacServiceAbstract () [inline, protected]

Default Constructor.

This constructor is protected to ensure the class is abstract.

Definition at line 31 of file FacServiceAbstract.hpp.

10.6.4 Member Function Documentation

10.6.4.1 void SIMCRS::FacServiceAbstract::clean ()

Destroyed all the object instantiated by this factory.

Definition at line 18 of file FacServiceAbstract.cpp.

References `_pool`.

Referenced by `~FacServiceAbstract()`.

10.6.5 Member Data Documentation

10.6.5.1 ServicePool_T SIMCRS::FacServiceAbstract::_pool [protected]

List of instantiated Business Objects

Definition at line 34 of file FacServiceAbstract.hpp.

Referenced by `clean()`, and `SIMCRS::FacSimcrsServiceContext::create()`.

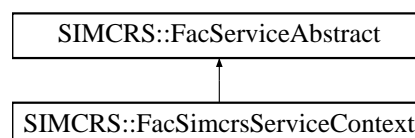
The documentation for this class was generated from the following files:

- `simcrs/factory/FacServiceAbstract.hpp`
- `simcrs/factory/FacServiceAbstract.cpp`

10.7 SIMCRS::FacSimcrsServiceContext Class Reference

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

Inheritance diagram for SIMCRS::FacSimcrsServiceContext::



Public Types

- typedef std::vector< [ServiceAbstract](#) * > [ServicePool_T](#)

Public Member Functions

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

Static Public Member Functions

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

Protected Member Functions

- [FacSimcrsServiceContext](#) ()

Protected Attributes

- [ServicePool_T _pool](#)

10.7.1 Detailed Description

Factory for Bucket.

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

10.7.2 Member Typedef Documentation

10.7.2.1 typedef std::vector<[ServiceAbstract](#)*> [SIMCRS::FacServiceAbstract::ServicePool_T](#)
[inherited]

Define the list (pool) of Service objects.

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

10.7.3 Constructor & Destructor Documentation

10.7.3.1 [SIMCRS::FacSimcrsServiceContext::~~FacSimcrsServiceContext](#) ()

Destructor.

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

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

10.7.3.2 SIMCRS::FacSimcrsServiceContext::FacSimcrsServiceContext () [inline, protected]

Default Constructor.

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

Definition at line 42 of file FacSimcrsServiceContext.hpp.

Referenced by instance().

10.7.4 Member Function Documentation

10.7.4.1 FacSimcrsServiceContext & SIMCRS::FacSimcrsServiceContext::instance () [static]

Provide the unique instance.

The singleton is instantiated when first used

Returns:

[FacSimcrsServiceContext](#)&

Definition at line 21 of file FacSimcrsServiceContext.cpp.

References [FacSimcrsServiceContext\(\)](#), [SIMCRS::FacSupervisor::instance\(\)](#), and [SIMCRS::FacSupervisor::registerServiceFactory\(\)](#).

10.7.4.2 SIMCRS_ServiceContext & SIMCRS::FacSimcrsServiceContext::create (const std::string & iTravelDatabaseName)

Create a new [SIMCRS_ServiceContext](#) object.

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

Returns:

[SIMCRS_ServiceContext](#)& The newly created object.

Definition at line 34 of file FacSimcrsServiceContext.cpp.

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

10.7.4.3 void SIMCRS::FacServiceAbstract::clean () [inherited]

Destroyed all the object instantiated by this factory.

Definition at line 18 of file FacServiceAbstract.cpp.

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

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

10.7.5 Member Data Documentation

10.7.5.1 ServicePool_T SIMCRS::FacServiceAbstract::_pool [protected, inherited]

List of instantiated Business Objects

Definition at line 34 of file FacServiceAbstract.hpp.

Referenced by SIMCRS::FacServiceAbstract::clean(), and create().

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

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

10.8 SIMCRS::FacSupervisor Class Reference

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

Public Types

- typedef std::vector< [FacBomAbstract *](#) > [BomFactoryPool_T](#)
- typedef std::vector< [FacServiceAbstract *](#) > [ServiceFactoryPool_T](#)

Public Member Functions

- void [registerBomFactory](#) ([FacBomAbstract *](#))
- void [registerServiceFactory](#) ([FacServiceAbstract *](#))
- void [cleanBomLayer](#) ()
- void [cleanServiceLayer](#) ()
- [~FacSupervisor](#) ()

Static Public Member Functions

- static [FacSupervisor](#) & [instance](#) ()
- static void [cleanFactory](#) ()

Protected Member Functions

- [FacSupervisor](#) ()
- [FacSupervisor](#) (const [FacSupervisor](#) &)

10.8.1 Detailed Description

Singleton class to register and clean all Factories.

Definition at line 17 of file FacSupervisor.hpp.

10.8.2 Member Typedef Documentation

10.8.2.1 typedef std::vector<[FacBomAbstract*](#)> [SIMCRS::FacSupervisor::BomFactoryPool_T](#)

Define the pool (list) of factories.

Definition at line 21 of file FacSupervisor.hpp.

10.8.2.2 typedef std::vector<FacServiceAbstract*> SIMCRS::FacSupervisor::ServiceFactory-Pool_T

Definition at line 22 of file FacSupervisor.hpp.

10.8.3 Constructor & Destructor Documentation

10.8.3.1 SIMCRS::FacSupervisor::~~FacSupervisor ()

Destructor

The static instance is deleted (and reset to NULL) by the static [cleanFactory\(\)](#) method.

Definition at line 41 of file FacSupervisor.cpp.

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

10.8.3.2 SIMCRS::FacSupervisor::FacSupervisor () [protected]

Default Constructor.

This constructor is protected to ensure the singleton pattern.

Definition at line 16 of file FacSupervisor.cpp.

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

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

Definition at line 66 of file FacSupervisor.hpp.

10.8.4 Member Function Documentation

10.8.4.1 FacSupervisor & SIMCRS::FacSupervisor::instance () [static]

Provides the unique instance.

The singleton is instantiated when first used.

Returns:

[FacSupervisor&](#)

Definition at line 20 of file FacSupervisor.cpp.

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

Referenced by [SIMCRS::FacSimcrsServiceContext::instance\(\)](#).

10.8.4.2 void SIMCRS::FacSupervisor::registerBomFactory (FacBomAbstract *)

Register a newly instantiated concrete factory for the Bom layer.

When a concrete Factory is firstly instantiated this factory have to register itself to the [FacSupervisor](#)

Parameters:

FacAbstract& the concrete Factory to register.

Definition at line 30 of file FacSupervisor.cpp.

10.8.4.3 void SIMCRS::FacSupervisor::registerServiceFactory (FacServiceAbstract *)

Register a newly instantiated concrete factory for the Service layer.

When a concrete Factory is firstly instantiated this factory have to register itself to the [FacSupervisor](#).

Parameters:

*FacServiceAbstract** the concrete Factory to register.

Definition at line 36 of file FacSupervisor.cpp.

Referenced by SIMCRS::FacSimcrsServiceContext::instance().

10.8.4.4 void SIMCRS::FacSupervisor::cleanBomLayer ()

Clean all created object.

Call the clean method of all the instantiated factories for the Bom layer.

Definition at line 47 of file FacSupervisor.cpp.

Referenced by cleanFactory(), and ~FacSupervisor().

10.8.4.5 void SIMCRS::FacSupervisor::cleanServiceLayer ()

Clean all Service created object.

Call the clean method of all the instantiated factories for the Service layer.

Definition at line 61 of file FacSupervisor.cpp.

Referenced by cleanFactory(), and ~FacSupervisor().

10.8.4.6 void SIMCRS::FacSupervisor::cleanFactory () [static]

Clean the static instance.

The singleton is deleted.

Definition at line 75 of file FacSupervisor.cpp.

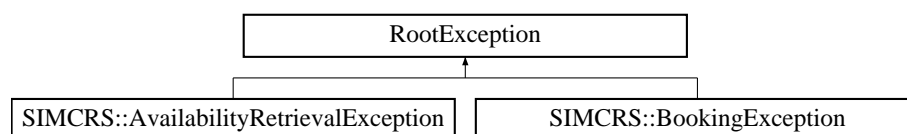
References cleanBomLayer(), and cleanServiceLayer().

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

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

10.9 RootException Class Reference

Inheritance diagram for RootException::



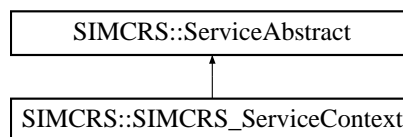
The documentation for this class was generated from the following file:

- [simcrs/SIMCRS_Types.hpp](#)

10.10 SIMCRS::ServiceAbstract Class Reference

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

Inheritance diagram for SIMCRS::ServiceAbstract::



Public Member Functions

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

Protected Member Functions

- [ServiceAbstract](#) ()

10.10.1 Detailed Description

Base class for the Service layer.

Definition at line 14 of file ServiceAbstract.hpp.

10.10.2 Constructor & Destructor Documentation

10.10.2.1 virtual SIMCRS::ServiceAbstract::~~ServiceAbstract () [inline, virtual]

Destructor.

Definition at line 18 of file ServiceAbstract.hpp.

10.10.2.2 SIMCRS::ServiceAbstract::ServiceAbstract () [inline, protected]

Protected Default Constructor to ensure this class is abstract.

Definition at line 30 of file ServiceAbstract.hpp.

10.10.3 Member Function Documentation

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

Dump a Business Object into an output stream.

Parameters:

ostream& the output stream.

Definition at line 22 of file ServiceAbstract.hpp.

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

Read a Business Object from an input stream.

Parameters:

istream& the input stream.

Definition at line 26 of file ServiceAbstract.hpp.

Referenced by operator>>().

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

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

10.11 SIMCRS::SIMCRS_Service Class Reference

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

Public Member Functions

- [SIMCRS_Service](#) (const stdair::BasLogParams &, const stdair::BasDBParams &, const [CRSCode_T](#) &)
- [SIMCRS_Service](#) (const stdair::BasLogParams &, const [CRSCode_T](#) &)
- [SIMCRS_Service](#) (stdair::STDAIR_ServicePtr_T, SEVMGR::SEVMGR_ServicePtr_T, const [CRSCode_T](#) &)
- void [parseAndLoad](#) (const stdair::ScheduleFilePath &, const stdair::ODFilePath &, const stdair::FRAT5FilePath &, const stdair::FFDisutilityFilePath &, const AIRRAC::YieldFilePath &, const SIMFQT::FareFilePath &)
- void [initSnapshotAndRMEvents](#) (const stdair::Date_T &iStartDate, const stdair::Date_T &iEndDate)
- [~SIMCRS_Service](#) ()
- stdair::TravelSolutionList_T [calculateSegmentPathList](#) (const stdair::BookingRequestStruct &)
- void [fareQuote](#) (const stdair::BookingRequestStruct &, stdair::TravelSolutionList_T &)
- void [calculateAvailability](#) (stdair::TravelSolutionList_T &)
- bool [sell](#) (const stdair::TravelSolutionStruct &, const stdair::PartySize_T &)
- void [takeSnapshots](#) (const stdair::SnapshotStruct &)
- bool [playCancellation](#) (const stdair::CancellationStruct &)
- void [optimise](#) (const stdair::RMEventStruct &)
- bool [sell](#) (const std::string &iSegmentDateKey, const stdair::ClassCode_T &, const stdair::PartySize_T &)
- void [buildSampleBom](#) ()
- void [clonePersistentBom](#) ()
- void [buildComplementaryLinks](#) (stdair::BomRoot &)
- void [buildSampleTravelSolutions](#) (stdair::TravelSolutionList_T &)

- stdair::BookingRequestStruct [buildSampleBookingRequest](#) (const bool isForCRS=false)
- std::string [jsonHandler](#) (const stdair::JSONString &) const
- std::string [csvDisplay](#) () const
- std::string [csvDisplay](#) (const stdair::TravelSolutionList_T &) const
- std::string [list](#) (const stdair::AirlineCode_T &iAirlineCode="all", const stdair::FlightNumber_T &iFlightNumber=0) const
- std::string [csvDisplay](#) (const stdair::AirlineCode_T &, const stdair::FlightNumber_T &, const stdair::Date_T &iDepartureDate) const

10.11.1 Detailed Description

Interface for the [SIMCRS](#) Services.

Definition at line 42 of file SIMCRS_Service.hpp.

10.11.2 Constructor & Destructor Documentation

10.11.2.1 SIMCRS::SIMCRS_Service::SIMCRS_Service (const stdair::BasLogParams &, const stdair::BasDBParams &, const [CRSCode_T](#) &)

Constructor.

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

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

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

Parameters:

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

const stdair::BasDBParams& Parameters for the database access.

const CRSCode_T& Code of the owner of the distribution system.

Definition at line 81 of file SIMCRS_Service.cpp.

10.11.2.2 SIMCRS::SIMCRS_Service::SIMCRS_Service (const stdair::BasLogParams &, const [CRSCode_T](#) &)

Constructor.

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

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

Parameters:

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

const CRSCode_T& Code of the owner of the distribution system.

Definition at line 51 of file SIMCRS_Service.cpp.

10.11.2.3 SIMCRS::SIMCRS_Service::SIMCRS_Service (stdair::STDAIR_ServicePtr_T, SEVMGR::SEVMGR_ServicePtr_T, const CRSCode_T &)

Constructor.

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

Moreover, as no reference on any output stream is given, it is assumed that the StdAir log service has already been initialised with the proper log output stream by some other methods in the calling chain (for instance, when the [SIMCRS_Service](#) is itself being initialised by another library service such as TVLSIM_Service).

Parameters:

stdair::STDAIR_ServicePtr_T Reference on the STDAIR service.

SEVMGR::SEVMGR_ServicePtr_T Reference on the SEVMGR_Service.

const stdair::RandomSeed_T& Seed for the random generation.

const CRSCode_T& Code of the owner of the distribution system.

Definition at line 113 of file SIMCRS_Service.cpp.

10.11.2.4 SIMCRS::SIMCRS_Service::~~SIMCRS_Service ()

Destructor.

Definition at line 144 of file SIMCRS_Service.cpp.

10.11.3 Member Function Documentation

10.11.3.1 void SIMCRS::SIMCRS_Service::parseAndLoad (const stdair::ScheduleFilePath &, const stdair::ODFilePath &, const stdair::FRAT5FilePath &, const stdair::FFDisutilityFilePath &, const AIRRAC::YieldFilePath &, const SIMFQT::FareFilePath &)

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

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

Parameters:

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

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

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

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

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

const SIMFQT::FareFilePath& Filename of the input fare file.

Definition at line 324 of file SIMCRS_Service.cpp.

References [buildComplementaryLinks\(\)](#), [clonePersistentBom\(\)](#), [SIMCRS::SIMCRS_ServiceContext::getAIRINV_Service\(\)](#), [SIMCRS::SIMCRS_ServiceContext::getAIRSCHED_Service\(\)](#), [SIMCRS::SIMCRS_ServiceContext::getOwnStdairServiceFlag\(\)](#), [SIMCRS::SIMCRS_ServiceContext::getSIMFQT_Service\(\)](#), and [SIMCRS::SIMCRS_ServiceContext::getSTDAIR_Service\(\)](#).

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

10.11.3.2 void SIMCRS::SIMCRS_Service::initSnapshotAndRMEvents (const stdair::Date_T & i-StartDate, const stdair::Date_T & iEndDate)

Initialise the snapshot and RM events for the inventories.

Parameters:

const stdair::Date_T& Start date of the simulation.

const stdair::Date_T& End date of the simulation.

Definition at line 635 of file SIMCRS_Service.cpp.

References SIMCRS::SIMCRS_ServiceContext::getAIRINV_Service().

10.11.3.3 stdair::TravelSolutionList_T SIMCRS::SIMCRS_Service::calculateSegmentPathList (const stdair::BookingRequestStruct &)

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

Definition at line 739 of file SIMCRS_Service.cpp.

References SIMCRS::SIMCRS_ServiceContext::display(), and SIMCRS::SIMCRS_ServiceContext::get-AIRSCHEM_Service().

Referenced by main().

10.11.3.4 void SIMCRS::SIMCRS_Service::fareQuote (const stdair::BookingRequestStruct &, stdair::TravelSolutionList_T &)

Calculate the fare of each travel solutions in the list.

Definition at line 775 of file SIMCRS_Service.cpp.

References SIMCRS::SIMCRS_ServiceContext::display(), and SIMCRS::SIMCRS_ServiceContext::get-SIMFQT_Service().

Referenced by main().

10.11.3.5 void SIMCRS::SIMCRS_Service::calculateAvailability (stdair::TravelSolutionList_T &)

Compute the availability for each travel solution in the list.

Definition at line 806 of file SIMCRS_Service.cpp.

References SIMCRS::SIMCRS_ServiceContext::display(), and SIMCRS::SIMCRS_ServiceContext::get-AIRINV_Service().

10.11.3.6 bool SIMCRS::SIMCRS_Service::sell (const stdair::TravelSolutionStruct &, const stdair::PartySize_T &)

Register a booking.

Definition at line 839 of file SIMCRS_Service.cpp.

References SIMCRS::SIMCRS_ServiceContext::display(), and SIMCRS::SIMCRS_ServiceContext::get-AIRINV_Service().

Referenced by main().

10.11.3.7 void SIMCRS::SIMCRS_Service::takeSnapshots (const stdair::SnapshotStruct &)

Take inventory snapshots.

Definition at line 925 of file SIMCRS_Service.cpp.

References SIMCRS::SIMCRS_ServiceContext::getAIRINV_Service().

10.11.3.8 bool SIMCRS::SIMCRS_Service::playCancellation (const stdair::CancellationStruct &)

Play cancellation.

Definition at line 886 of file SIMCRS_Service.cpp.

References SIMCRS::SIMCRS_ServiceContext::display(), and SIMCRS::SIMCRS_ServiceContext::getAIRINV_Service().

10.11.3.9 void SIMCRS::SIMCRS_Service::optimise (const stdair::RMEventStruct &)

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

Definition at line 944 of file SIMCRS_Service.cpp.

References SIMCRS::SIMCRS_ServiceContext::getAIRINV_Service().

10.11.3.10 bool SIMCRS::SIMCRS_Service::sell (const std::string & *iSegmentDateKey*, const stdair::ClassCode_T &, const stdair::PartySize_T &)

Register a booking.

Parameters:

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

const stdair::ClassCode_T& Class code where the sale is made

const stdair::PartySize_T& Party size

Returns:

bool Whether or not the sale was successful

Definition at line 593 of file SIMCRS_Service.cpp.

References SIMCRS::SIMCRS_ServiceContext::getAIRINV_Service().

10.11.3.11 void SIMCRS::SIMCRS_Service::buildSampleBom ()

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

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

See also:

AIRINV::AIRINV_Master_Service and stdair::CmdBomManager for more details.

Definition at line 402 of file SIMCRS_Service.cpp.

References buildComplementaryLinks(), clonePersistentBom(), SIMCRS::SIMCRS_ServiceContext::getAIRINV_Service(), SIMCRS::SIMCRS_ServiceContext::getAIRSCHED_Service(),

SIMCRS::SIMCRS_ServiceContext::getOwnStdairServiceFlag(), SIMCRS::SIMCRS_ServiceContext::getSIMFQT_Service(), and SIMCRS::SIMCRS_ServiceContext::getSTDAIR_Service().

Referenced by main().

10.11.3.12 void SIMCRS::SIMCRS_Service::clonePersistentBom ()

Clone the persistent BOM object.

Definition at line 481 of file SIMCRS_Service.cpp.

References buildComplementaryLinks(), SIMCRS::SIMCRS_ServiceContext::getAIRINV_Service(), SIMCRS::SIMCRS_ServiceContext::getAIRSCHED_Service(), SIMCRS::SIMCRS_ServiceContext::getOwnStdairServiceFlag(), SIMCRS::SIMCRS_ServiceContext::getSIMFQT_Service(), and SIMCRS::SIMCRS_ServiceContext::getSTDAIR_Service().

Referenced by buildSampleBom(), and parseAndLoad().

10.11.3.13 void SIMCRS::SIMCRS_Service::buildComplementaryLinks (stdair::BomRoot &)

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

Note:

Do nothing for now.

Definition at line 548 of file SIMCRS_Service.cpp.

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

10.11.3.14 void SIMCRS::SIMCRS_Service::buildSampleTravelSolutions (stdair::TravelSolutionList_T &)

Build a sample list of travel solutions.

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

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

See also:

stdair::CmdBomManager for more details.

Parameters:

TravelSolutionList_T& Sample list of travel solution structures. It should be given empty. It is altered with the returned sample.

Definition at line 554 of file SIMCRS_Service.cpp.

References SIMCRS::SIMCRS_ServiceContext::getSTDAIR_Service().

10.11.3.15 stdair::BookingRequestStruct SIMCRS::SIMCRS_Service::buildSampleBookingRequest (const bool *isForCRS* = false)

Build a sample booking request structure.

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

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

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

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

See also:

stdair::CmdBomManager for more details.

Parameters:

const bool *isForCRS* Whether the sample booking request is for CRS.

Returns:

BookingRequestStruct& Sample booking request structure.

Definition at line 574 of file SIMCRS_Service.cpp.

References SIMCRS::SIMCRS_ServiceContext::getSTDAIR_Service().

Referenced by main().

10.11.3.16 std::string SIMCRS::SIMCRS_Service::jsonHandler (const stdair::JSONString &) const

Dispatch the JSon command string to the AirInv service. (Only AirInv has json export commands for now).

Parameters:

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

Returns:

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

Definition at line 615 of file SIMCRS_Service.cpp.

References SIMCRS::SIMCRS_ServiceContext::getAIRINV_Service().

10.11.3.17 std::string SIMCRS::SIMCRS_Service::csvDisplay () const

Recursively display (dump in the returned string) the objects of the BOM tree.

Returns:

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

Definition at line 654 of file SIMCRS_Service.cpp.

References SIMCRS::SIMCRS_ServiceContext::getSTDAIR_Service().

Referenced by main().

10.11.3.18 std::string SIMCRS::SIMCRS_Service::csvDisplay (const stdair::TravelSolutionList_T &) const

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

Returns:

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

Definition at line 675 of file SIMCRS_Service.cpp.

References SIMCRS::SIMCRS_ServiceContext::getSTDAIR_Service().

10.11.3.19 std::string SIMCRS::SIMCRS_Service::list (const stdair::AirlineCode_T & iAirlineCode = "all", const stdair::FlightNumber_T & iFlightNumber = 0) const

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

Parameters:

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

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

Returns:

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

Definition at line 695 of file SIMCRS_Service.cpp.

References SIMCRS::SIMCRS_ServiceContext::getAIRINV_Service().

10.11.3.20 std::string SIMCRS::SIMCRS_Service::csvDisplay (const stdair::AirlineCode_T &, const stdair::FlightNumber_T &, const stdair::Date_T & *iDepartureDate*) const

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

Parameters:

- const** stdair::AirlineCode_T& Airline code of the flight to display
- const** stdair::FlightNumber_T& Flight number of the flight to display.
- const** stdair::Date_T& Departure date of the flight to display.

Returns:

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

Definition at line 716 of file SIMCRS_Service.cpp.

References SIMCRS::SIMCRS_ServiceContext::getAIRINV_Service().

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

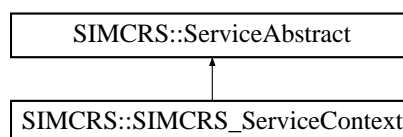
- [simcrs/SIMCRS_Service.hpp](#)
- [simcrs/service/SIMCRS_Service.cpp](#)

10.12 SIMCRS::SIMCRS_ServiceContext Class Reference

Class holding the context of the Simcrs services.

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

Inheritance diagram for SIMCRS::SIMCRS_ServiceContext::



Public Member Functions

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

Friends

- class [SIMCRS_Service](#)
- class [FacSimcrsServiceContext](#)

10.12.1 Detailed Description

Class holding the context of the Simcrs services.

Definition at line 32 of file SIMCRS_ServiceContext.hpp.

10.12.2 Member Function Documentation

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

Dump a Business Object into an output stream.

Parameters:

ostream& the output stream.

Definition at line 22 of file ServiceAbstract.hpp.

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

Read a Business Object from an input stream.

Parameters:

istream& the input stream.

Definition at line 26 of file ServiceAbstract.hpp.

Referenced by operator>>().

10.12.3 Friends And Related Function Documentation

10.12.3.1 `friend class SIMCRS_Service` `[friend]`

The [SIMCRS_Service](#) class should be the sole class to get access to ServiceContext content: general users do not want to bother with a context interface.

Definition at line 38 of file SIMCRS_ServiceContext.hpp.

10.12.3.2 `friend class FacSimcrsServiceContext` `[friend]`

Definition at line 39 of file SIMCRS_ServiceContext.hpp.

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

- [simcrs/service/SIMCRS_ServiceContext.hpp](#)
- [simcrs/service/SIMCRS_ServiceContext.cpp](#)

11 SimCRS File Documentation

11.1 doc/local/authors.doc File Reference

11.2 doc/local/codingrules.doc File Reference

11.3 doc/local/copyright.doc File Reference

11.4 doc/local/documentation.doc File Reference

11.5 doc/local/features.doc File Reference

11.6 doc/local/help_wanted.doc File Reference

11.7 doc/local/howto_release.doc File Reference

11.8 doc/local/index.doc File Reference

11.9 doc/local/installation.doc File Reference

11.10 doc/local/linking.doc File Reference

11.11 doc/local/test.doc File Reference

11.12 doc/local/users_guide.doc File Reference

11.13 doc/local/verification.doc File Reference

11.14 doc/tutorial/tutorial.doc File Reference

11.15 simcrs/basic/BasConst.cpp File Reference

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

Namespaces

- namespace [SIMCRS](#)

Variables

- const std::string [SIMCRS::DEFAULT_CRS_CODE](#) = "1S"

11.16 simcrs/basic/BasConst_General.hpp File Reference

Namespaces

- namespace [SIMCRS](#)

11.17 simcrs/basic/BasConst_SIMCRS_Service.hpp File Reference

```
#include <string>
```

Namespaces

- namespace [SIMCRS](#)

Variables

- const std::string [SIMCRS::DEFAULT_CRIS_CODE](#)

11.18 simcrs/batches/simcrs.cpp File Reference

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

Functions

- const std::string [K_SIMCRS_DEFAULT_LOG_FILENAME](#) ("simcrs.log")
- const std::string [K_SIMCRS_DEFAULT_SCHEDULE_INPUT_FILENAME](#) (STDAIR_SAMPLE_DIR"/schedule01.csv")
- const std::string [K_SIMCRS_DEFAULT_OND_INPUT_FILENAME](#) (STDAIR_SAMPLE_DIR"/ond01.csv")
- const std::string [K_SIMCRS_DEFAULT_FRAT5_INPUT_FILENAME](#) (STDAIR_SAMPLE_DIR"/frat5.csv")
- const std::string [K_SIMCRS_DEFAULT_FF_DISUTILITY_INPUT_FILENAME](#) (STDAIR_SAMPLE_DIR"/ffDisutility.csv")
- const std::string [K_SIMCRS_DEFAULT_YIELD_INPUT_FILENAME](#) (STDAIR_SAMPLE_DIR"/yieldstore01.csv")
- const std::string [K_SIMCRS_DEFAULT_FARE_INPUT_FILENAME](#) (STDAIR_SAMPLE_DIR"/fare01.csv")

- const std::string [K_SIMCRS_DEFAULT_DB_USER](#) ("dsim")
- const std::string [K_SIMCRS_DEFAULT_DB_PASSWD](#) ("dsim")
- const std::string [K_SIMCRS_DEFAULT_DB_DBNAME](#) ("sim_dsim")
- const std::string [K_SIMCRS_DEFAULT_DB_HOST](#) ("localhost")
- const std::string [K_SIMCRS_DEFAULT_DB_PORT](#) ("3306")
- template<class T> std::ostream & [operator<<](#) (std::ostream &os, const std::vector< T > &v)
- int [readConfiguration](#) (int argc, char *argv[], bool &ioIsBuiltin, stdair::Filename_T &ioScheduleInputFilename, stdair::Filename_T &ioOnDInputFilename, stdair::Filename_T &ioFRAT5Filename, stdair::Filename_T &ioFFDisutilityFilename, stdair::Filename_T &ioYieldInputFilename, stdair::Filename_T &ioFareInputFilename, stdair::Filename_T &ioLogFilename, std::string &ioDBUser, std::string &ioDBPasswd, std::string &ioDBHost, std::string &ioDBPort, std::string &ioDBDBName)
- int [main](#) (int argc, char *argv[])

Variables

- const bool [K_SIMCRS_DEFAULT_BUILT_IN_INPUT](#) = false
- const int [K_SIMCRS_EARLY_RETURN_STATUS](#) = 99

11.18.1 Function Documentation

11.18.1.1 const std::string [K_SIMCRS_DEFAULT_LOG_FILENAME](#) ("simcrs.log")

Default name and location for the log file.

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

11.18.1.2 const std::string [K_SIMCRS_DEFAULT_SCHEDULE_INPUT_FILENAME](#) (STDAIR_SAMPLE_DIR"/schedule01.csv")

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

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

11.18.1.3 const std::string [K_SIMCRS_DEFAULT_OND_INPUT_FILENAME](#) (STDAIR_SAMPLE_DIR"/ond01.csv")

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

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

11.18.1.4 const std::string [K_SIMCRS_DEFAULT_FRAT5_INPUT_FILENAME](#) (STDAIR_SAMPLE_DIR"/frat5.csv")

FRAT5 curve input file name

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

11.18.1.5 const std::string [K_SIMCRS_DEFAULT_FF_DISUTILITY_INPUT_FILENAME](#) (STDAIR_SAMPLE_DIR"/ffDisutility.csv")

Fare family disutility curve input file name

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

11.18.1.6 `const std::string K_SIMCRS_DEFAULT_YIELD_INPUT_FILENAME (STDAIR - SAMPLE_DIR"/yieldstore01.csv")`

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

Referenced by `readConfiguration()`.

11.18.1.7 `const std::string K_SIMCRS_DEFAULT_FARE_INPUT_FILENAME (STDAIR - SAMPLE_DIR"/fare01.csv")`

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

Referenced by `readConfiguration()`.

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

Default name and location for the MySQL database.

Referenced by `readConfiguration()`.

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

Referenced by `readConfiguration()`.

11.18.1.10 `const std::string K_SIMCRS_DEFAULT_DB_DBNAME ("sim_dsim")`

Referenced by `readConfiguration()`.

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

Referenced by `readConfiguration()`.

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

Referenced by `readConfiguration()`.

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

Definition at line 80 of file `simcrs.cpp`.

11.18.1.14 `int readConfiguration (int argc, char * argv[], bool & ioIsBuiltin, stdair::Filename_T & ioScheduleInputFilename, stdair::Filename_T & ioOnDInputFilename, stdair::Filename_T & ioFRAT5Filename, stdair::Filename_T & ioFFDisutilityFilename, stdair::Filename_T & ioYieldInputFilename, stdair::Filename_T & ioFareInputFilename, stdair::Filename_T & ioLogFilename, std::string & ioDBUser, std::string & ioDBPasswd, std::string & ioDBHost, std::string & ioDBPort, std::string & ioDBDBName)`

Read and parse the command line options.

Definition at line 90 of file `simcrs.cpp`.

References `K_SIMCRS_DEFAULT_BUILT_IN_INPUT`, `K_SIMCRS_DEFAULT_DB_DBNAME()`, `K_SIMCRS_DEFAULT_DB_HOST()`, `K_SIMCRS_DEFAULT_DB_PASSWD()`, `K_SIMCRS_DEFAULT_DB_PORT()`, `K_SIMCRS_DEFAULT_DB_USER()`, `K_SIMCRS_DEFAULT_FARE_`

INPUT_FILENAME(), K_SIMCRS_DEFAULT_FF_DISUTILITY_INPUT_FILENAME(), K_SIMCRS_DEFAULT_FRAT5_INPUT_FILENAME(), K_SIMCRS_DEFAULT_LOG_FILENAME(), K_SIMCRS_DEFAULT_OND_INPUT_FILENAME(), K_SIMCRS_DEFAULT_SCHEDULE_INPUT_FILENAME(), K_SIMCRS_DEFAULT_YIELD_INPUT_FILENAME(), K_SIMCRS_EARLY_RETURN_STATUS, PACKAGE_NAME, PACKAGE_VERSION, and PREFIXDIR.

Referenced by main().

11.18.1.15 int main (int argc, char * argv[])

Definition at line 313 of file simcrs.cpp.

References SIMCRS::SIMCRS_Service::buildSampleBom(), SIMCRS::SIMCRS_Service::buildSampleBookingRequest(), SIMCRS::SIMCRS_Service::calculateSegmentPathList(), SIMCRS::SIMCRS_Service::csvDisplay(), SIMCRS::SIMCRS_Service::fareQuote(), K_SIMCRS_EARLY_RETURN_STATUS, SIMCRS::SIMCRS_Service::parseAndLoad(), readConfiguration(), and SIMCRS::SIMCRS_Service::sell().

11.18.2 Variable Documentation

11.18.2.1 const bool K_SIMCRS_DEFAULT_BUILT_IN_INPUT = false

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

Definition at line 67 of file simcrs.cpp.

Referenced by readConfiguration().

11.18.2.2 const int K_SIMCRS_EARLY_RETURN_STATUS = 99

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

Definition at line 87 of file simcrs.cpp.

Referenced by main(), and readConfiguration().

11.19 simcrs/bom/BomAbstract.cpp File Reference

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

Namespaces

- namespace [SIMCRS](#)

11.20 simcrs/bom/BomAbstract.hpp File Reference

```
#include <iosfwd>
```

```
#include <string>
```

Namespaces

- namespace [SIMCRS](#)

Classes

- class [SIMCRS::BomAbstract](#)

Functions

- `template<class charT, class traits> std::basic_ostream< charT, traits > & operator<< (std::basic_ostream< charT, traits > &ioOut, const SIMCRS::BomAbstract &iBom)`
- `template<class charT, class traits> std::basic_istream< charT, traits > & operator>> (std::basic_istream< charT, traits > &ioIn, SIMCRS::BomAbstract &iBom)`

11.20.1 Function Documentation

11.20.1.1 `template<class charT, class traits> std::basic_ostream<charT, traits>& operator<< (std::basic_ostream< charT, traits > &ioOut, const SIMCRS::BomAbstract &iBom) [inline]`

Piece of code given by Nicolai M. Josuttis, Section 13.12.1 "Implementing Output Operators" (p653) of his book "The C++ Standard Library: A Tutorial and Reference", published by Addison-Wesley.

Definition at line 56 of file BomAbstract.hpp.

11.20.1.2 `template<class charT, class traits> std::basic_istream<charT, traits>& operator>> (std::basic_istream< charT, traits > &ioIn, SIMCRS::BomAbstract &iBom) [inline]`

Piece of code given by Nicolai M. Josuttis, Section 13.12.1 "Implementing Output Operators" (pp655-657) of his book "The C++ Standard Library: A Tutorial and Reference", published by Addison-Wesley.

Definition at line 84 of file BomAbstract.hpp.

References [SIMCRS::BomAbstract::fromStream\(\)](#).

11.21 simcrs/command/DistributionManager.cpp File Reference

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

Namespaces

- namespace [SIMCRS](#)

11.22 simcrs/command/DistributionManager.hpp File Reference

```
#include <stdair/stdair_basic_types.hpp>
```

```
#include <stdair/bom/TravelSolutionTypes.hpp>
#include <airinv/AIRINV_Types.hpp>
#include <simcrs/SIMCRS_Types.hpp>
```

Namespaces

- namespace [stdair](#)
- namespace [AIRINV](#)
- namespace [SIMCRS](#)

Classes

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

11.23 simcrs/config/simcrs-paths.hpp.in File Reference

Defines

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

11.23.1 Define Documentation

11.23.1.1 #define PACKAGE "@PACKAGE@"

Definition at line 4 of file `simcrs-paths.hpp.in`.

11.23.1.2 #define PACKAGE_NAME "@PACKAGE_NAME@"

Definition at line 5 of file `simcrs-paths.hpp.in`.

Referenced by `readConfiguration()`.

11.23.1.3 #define PACKAGE_VERSION "@PACKAGE_VERSION@"

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

Referenced by readConfiguration().

11.23.1.4 #define PREFIXDIR "@prefix@"

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

Referenced by readConfiguration().

11.23.1.5 #define EXEC_PREFIX "@exec_prefix@"

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

11.23.1.6 #define BINDIR "@bindir@"

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

11.23.1.7 #define LIBDIR "@libdir@"

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

11.23.1.8 #define LIBEXECDIR "@libexecdir@"

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

11.23.1.9 #define SBINDIR "@sbindir@"

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

11.23.1.10 #define SYSCONFDIR "@sysconfdir@"

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

11.23.1.11 #define INCLUDEDIR "@includedir@"

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

11.23.1.12 #define DATAROOTDIR "@datarootdir@"

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

11.23.1.13 #define DATADIR "@datadir@"

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

11.23.1.14 #define DOCDIR "@docdir@"

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

11.23.1.15 #define MANDIR "@mandir@"

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

11.23.1.16 #define INFODIR "@infodir@"

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

11.23.1.17 #define HTMLDIR "@htmldir@"

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

11.23.1.18 #define PDFDIR "@pdfdir@"

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

11.23.1.19 #define STDAIR_SAMPLE_DIR "@sampledir@"

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

11.24 simcrs/factory/FacBomAbstract.cpp File Reference

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

Namespaces

- namespace [SIMCRS](#)

11.25 simcrs/factory/FacBomAbstract.hpp File Reference

```
#include <string>
#include <vector>
```

Namespaces

- namespace [SIMCRS](#)

Classes

- class [SIMCRS::FacBomAbstract](#)

11.26 simcrs/factory/FacServiceAbstract.cpp File Reference

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

Namespaces

- namespace [SIMCRS](#)

11.27 simcrs/factory/FacServiceAbstract.hpp File Reference

```
#include <vector>
```

Namespaces

- namespace [SIMCRS](#)

Classes

- class [SIMCRS::FacServiceAbstract](#)

11.28 simcrs/factory/FacSimcrsServiceContext.cpp File Reference

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

Namespaces

- namespace [SIMCRS](#)

11.29 simcrs/factory/FacSimcrsServiceContext.hpp File Reference

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

Namespaces

- namespace [SIMCRS](#)

Classes

- class [SIMCRS::FacSimcrsServiceContext](#)

11.30 `simcrs/factory/FacSupervisor.cpp` File Reference

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

Namespaces

- namespace [SIMCRS](#)

11.31 `simcrs/factory/FacSupervisor.hpp` File Reference

```
#include <vector>
```

Namespaces

- namespace [SIMCRS](#)

Classes

- class [SIMCRS::FacSupervisor](#)

11.32 `simcrs/service/ServiceAbstract.cpp` File Reference

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

Namespaces

- namespace [SIMCRS](#)

11.33 `simcrs/service/ServiceAbstract.hpp` File Reference

```
#include <iosfwd>
```

Namespaces

- namespace [SIMCRS](#)

Classes

- class [SIMCRS::ServiceAbstract](#)

Functions

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

11.33.1 Function Documentation

11.33.1.1 `template<class charT, class traits> std::basic_ostream<charT, traits>& operator<< (std::basic_ostream< charT, traits > & ioOut, const SIMCRS::ServiceAbstract & iService) [inline]`

Piece of code given by Nicolai M. Josuttis, Section 13.12.1 "Implementing Output Operators" (p653) of his book "The C++ Standard Library: A Tutorial and Reference", published by Addison-Wesley.

Definition at line 42 of file ServiceAbstract.hpp.

11.33.1.2 `template<class charT, class traits> std::basic_istream<charT, traits>& operator>> (std::basic_istream< charT, traits > & ioIn, SIMCRS::ServiceAbstract & ioService) [inline]`

Piece of code given by Nicolai M. Josuttis, Section 13.12.1 "Implementing Output Operators" (pp655-657) of his book "The C++ Standard Library: A Tutorial and Reference", published by Addison-Wesley.

Definition at line 70 of file ServiceAbstract.hpp.

References `SIMCRS::ServiceAbstract::fromStream()`.

11.34 simcrs/service/SIMCRS_Service.cpp File Reference

```
#include <cassert>
#include <sstream>
#include <boost/make_shared.hpp>
#include <stdair/stdair_exceptions.hpp>
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_json.hpp>
#include <stdair/basic/BasChronometer.hpp>
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/bom/BomManager.hpp>
#include <stdair/bom/BookingRequestStruct.hpp>
#include <stdair/bom/TravelSolutionStruct.hpp>
#include <stdair/bom/CancellationStruct.hpp>
#include <stdair/bom/BomRoot.hpp>
#include <stdair/bom/Inventory.hpp>
#include <stdair/service/Logger.hpp>
#include <stdair/STDAIR_Service.hpp>
```

```
#include <sevmgr/SEVMGR_Service.hpp>
#include <airinv/AIRINV_Master_Service.hpp>
#include <airsched/AIRSCHEM_Service.hpp>
#include <simfqt/SIMFQT_Service.hpp>
#include <simcrs/basic/BasConst_SIMCRS_Service.hpp>
#include <simcrs/command/DistributionManager.hpp>
#include <simcrs/factory/FacSimcrsServiceContext.hpp>
#include <simcrs/service/SIMCRS_ServiceContext.hpp>
#include <simcrs/SIMCRS_Service.hpp>
```

Namespaces

- namespace [SIMCRS](#)

11.35 `simcrs/service/SIMCRS_ServiceContext.cpp` File Reference

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

Namespaces

- namespace [SIMCRS](#)

11.36 `simcrs/service/SIMCRS_ServiceContext.hpp` File Reference

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


Namespaces

- namespace [SIMCRS](#)

Classes

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

11.37 `simcrs/SIMCRS_Service.hpp` File Reference

```
#include <stdair/stdair_basic_types.hpp>
#include <stdair/stdair_file.hpp>
#include <stdair/stdair_service_types.hpp>
#include <stdair/bom/TravelSolutionTypes.hpp>
#include <sevmgr/SEVMGR_Types.hpp>
#include <simfqt/SIMFQT_Types.hpp>
#include <airrac/AIRRAC_Types.hpp>
#include <simcrs/SIMCRS_Types.hpp>
```

Namespaces

- namespace [stdair](#)
- namespace [SIMCRS](#)

Classes

- class [SIMCRS::SIMCRS_Service](#)

11.38 `simcrs/SIMCRS_Types.hpp` File Reference

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

Namespaces

- namespace [SIMCRS](#)

Classes

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

Typedefs

- typedef std::string [SIMCRS::CRSCode_T](#)
- typedef boost::shared_ptr< SIMCRS_Service > [SIMCRS::SIMCRS_ServicePtr_T](#)

11.39 test/simcrs/CRSTestSuite.cpp File Reference

12 SimCRS Page Documentation

12.1 People

12.1.1 Project Admins

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

12.1.2 Developers

- Anh Quan Nguyen <quannaus@users.sourceforge.net> ([N](#))
- Denis Arnaud <denis_arnaud@users.sourceforge.net> ([N](#))
- Son Nguyen Kim <snguyenkim@users.sourceforge.net>
- Nicolas Bondoux <nbondoux@users.sourceforge.net> ([N](#))

12.1.3 Retired Developers

- Patrick Grandjean <pgrandjean@users.sourceforge.net> ([N](#))
- Ngoc-Thach Hoang <hoangngocthach@users.sourceforge.net> ([N](#))

12.1.4 Contributors

- Emmanuel Bastien <ebastien@users.sourceforge.net> ([N](#))
- Christophe Lacombe <ddtof@users.sourceforge.net> ([N](#))

12.1.5 Distribution Maintainers

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

Note:

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

12.2 Coding Rules

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

12.2.1 Default Naming Rules for Variables

Variables names follow Java naming conventions. Examples:

- `lNumberOfPassengers`
- `lSeatAvailability`

12.2.2 Default Naming Rules for Functions

Function names follow Java naming conventions. Example:

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

12.2.3 Default Naming Rules for Classes and Structures

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

- `MyClassName`
- `MyStructName`

12.2.4 Default Naming Rules for Files

Files are named after the C++ class names.

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

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

12.2.5 Default Functionality of Classes

All classes that are configured by input parameters should include:

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

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

12.3 Copyright and License

12.3.1 GNU LESSER GENERAL PUBLIC LICENSE

12.3.1.1 Version 2.1, February 1999

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

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

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

12.3.2 Preamble

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

12.3.3 TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

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

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

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

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

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

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate

copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

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

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

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

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

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

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

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

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

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

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

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

If distribution of object code is made by offering access to copy from a designated place, then offering

equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

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

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

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

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

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

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

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to

be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

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

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

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

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

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

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

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

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

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

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

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add

an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

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

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

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

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

12.3.3.2 END OF TERMS AND CONDITIONS

12.3.4 How to Apply These Terms to Your New Programs

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

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

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

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

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

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

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

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

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

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

That's all there is to it!

Source

12.4 Documentation Rules

12.4.1 General Rules

All classes in SimCRS should be properly documented with Doxygen comments in include (.hpp) files. Source (.cpp) files should be documented according to a normal standard for well documented C++ code.

An example of how the interface of a class shall be documented in SimCRS is shown here:

```
/*!
 * \brief Brief description of MyClass here
 *
 * Detailed description of MyClass here. With example code if needed.
 */
class MyClass {
public:
    /*! Default constructor
    MyClass(void) { setup_done = false; }

    /*!
    * \brief Constructor that initializes the class with parameters
    *
    * Detailed description of the constructor here if needed
    *
    * \param[in] param1 Description of \a param1 here
    * \param[in] param2 Description of \a param2 here
    */
    MyClass(TYPE1 param1, TYPE2 param2) { setup(param1, param2); }

    /*!
    * \brief Setup function for MyClass
    *
    * Detailed description of the setup function here if needed
```

```

*
* \param[in] param1 Description of \a param1 here
* \param[in] param2 Description of \a param2 here
*/
void setup(TYPE1 param1, TYPE2 param2);

/*!
* \brief Brief description of memberFunction1
*
* Detailed description of memberFunction1 here if needed
*
* \param[in]      param1 Description of \a param1 here
* \param[in]      param2 Description of \a param2 here
* \param[in,out] param3 Description of \a param3 here
* \return Description of the return value here
*/
TYPE4 memberFunction1(TYPE1 param1, TYPE2 param2, TYPE3 &param3);

private:

    bool _setUpDone;          /*!< Variable that checks if the class is properly
                               initialized with parameters */
    TYPE1 _privateVariable1; /*!< Short description of _privateVariable1 here
    TYPE2 _privateVariable2; /*!< Short description of _privateVariable2 here
};

```

12.4.2 File Header

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

```

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

```

12.4.3 Grouping Various Parts

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

```

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

```

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

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

12.5 Main features

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

12.5.1 Network generation

- Network/graph generation

12.5.2 Inventory generation

- Inventory generation

12.5.3 Finding travel solutions

- Matching of travel solutions with user requests

12.5.4 Distributed inventories

- Inventory independent partitions
- MPI-based distribution

12.5.5 Other features

- CSV input file parsing
- Memory handling

12.6 Make a Difference

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

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

- Start using SimCRS

- Tell your friends about SimCRS and help them to get started using it
- If you find a bug, report it to us. Without your help we can never hope to produce a bug free code.
- Help us to improve the documentation by providing information about documentation bugs
- Answer support requests in the SimCRS discussion forums on SourceForge. If you know the answer to a question, help others to overcome their SimCRS problems.
- Help us to improve our algorithms. If you know of a better way (e.g. that is faster or requires less memory) to implement some of our algorithms, then let us know.
- Help us to port SimCRS to new platforms. If you manage to compile SimCRS on a new platform, then tell us how you did it.
- Send us your code. If you have a good SimCRS compatible code, which you can release under the LGPL, and you think it should be included in SimCRS, then send it to us.
- Become an SimCRS developer. Send us an e-mail and tell what you can do for SimCRS.

12.7 Make a new release

12.7.1 Introduction

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

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

12.7.2 Initialisation

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

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

12.7.3 Release branch maintenance

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

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

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

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

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

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

12.7.4 Commit and publish the release branch

Commit the new release:

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

12.7.5 Create distribution packages

Create the distribution packages using the following command:

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

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

12.7.6 Upload the HTML documentation to SourceForge

In order to update the Web site files, either:

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

```
cd ~/dev/sim/simcrsgit/build
git checkout releases
rsync -aiv ${INSTALL_BASEDIR}/simcrs-0.5.0/share/doc/simcrs-0.5.0/html/ \
  your_sf_user,simcrs@web.sourceforge.net:htdocs/
```

where `-aiv` options mean:

- `-a`: archive/mirror mode; equals `-rlptgoD` (no `-H`, `-A`, `-X`)

- `-v`: increase verbosity
 - `-i`: output a change-summary for all updates
 - Note the trailing slashes (/) at the end of both the source and target directories. It means that the content of the source directory (`doc/html`), rather than the directory itself, has to be copied into the content of the target directory.
- or use the [SourceForge Shell service](#).

12.7.7 Generate the RPM packages

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

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

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

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

12.7.8 Update distributed change log

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

12.7.9 Create the binary package, including the documentation

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

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

The output binary package will be named, for instance, `simcrs-0.5.0-Linux.tar.bz2`. That package contains both the HTML and PDF documentation. The binary package contains also the executables and shared libraries, as well as C++ header files, but all of those do not interest us for now.

12.7.10 Upload the files to SourceForge

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

12.7.11 Make a new post

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

12.7.12 Send an email on the announcement mailing-list

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

12.8 Installation

12.8.1 Table of Contents

- [Fedora/RedHat Linux distributions](#)
- [SimCRS Requirements](#)
- [Basic Installation](#)
- [Compilers and Options](#)
- [Compiling For Multiple Architectures](#)
- [Installation Names](#)
- [Optional Features](#)
- [Particular systems](#)
- [Specifying the System Type](#)
- [Sharing Defaults](#)
- [Defining Variables](#)
- [‘cmake’ Invocation](#)

12.8.2 Fedora/RedHat Linux distributions

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

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

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

12.8.3 SimCRS Requirements

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

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

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

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

12.8.4 Basic Installation

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

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

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

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

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

The simplest way to compile this package is:

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

12.8.5 Compilers and Options

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

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

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

See also:

[Defining Variables](#) for more details.

12.8.6 Compiling For Multiple Architectures

You can compile the package for more than one kind of computer at the same time, by placing the object files for each architecture in their own directory. To do this, you can use GNU `'make'`. `'cd'` to the directory where you want the object files and executables to go and

run the 'configure' script. 'configure' automatically checks for the source code in the directory that 'configure' is in and in '..'. This is known as a "VPATH" build.

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

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

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

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

12.8.7 Installation Names

By default, 'make install' installs the package's commands under '/usr/local/bin', include files under '/usr/local/include', etc. You can specify an installation prefix other than '/usr/local' by giving 'configure' the option '-prefix=PREFIX', where PREFIX must be an absolute file name.

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

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

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

The first method involves providing an override variable for each affected directory. For example, 'make install prefix=/alternate/directory' will choose an alternate location for all directory configuration variables that were expressed in terms of '\${prefix}'. Any directories that were specified during 'configure',

but not in terms of `'${prefix}'`, must each be overridden at install time for the entire installation to be relocated. The approach of makefile variable overrides for each directory variable is required by the GNU Coding Standards, and ideally causes no recompilation. However, some platforms have known limitations with the semantics of shared libraries that end up requiring recompilation when using this method, particularly noticeable in packages that use GNU Libtool.

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

12.8.8 Optional Features

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

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

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

Some packages offer the ability to configure how verbose the execution of `'make'` will be. For these packages, running `'./configure -enable-silent-rules'` sets the default to minimal output, which can be overridden with `'make V=1'`; while running `'./configure -disable-silent-rules'` sets the default to verbose, which can be overridden with `'make V=0'`.

12.8.9 Particular systems

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

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

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

On OSF/1 a.k.a. Tru64, some versions of the default C compiler cannot parse its `'<wchar.h>'` header file. The option `'-nodtk'` can be used as

a workaround. If GNU CC is not installed, it is therefore recommended to try

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

and if that doesn't work, try

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

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

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

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

12.8.10 Specifying the System Type

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

where SYSTEM can have one of these forms:

- OS
- KERNEL-OS

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

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

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

12.8.11 Sharing Defaults

If you want to set default values for `'configure'` scripts to share, you can create a site shell script called `'config.site'` that gives default values for variables like `'CC'`, `'cache_file'`, and `'prefix'`.

'configure' looks for 'PREFIX/share/config.site' if it exists, then 'PREFIX/etc/config.site' if it exists. Or, you can set the 'CONFIG_SITE' environment variable to the location of the site script. A warning: not all 'configure' scripts look for a site script.

12.8.12 Defining Variables

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

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

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

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

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

12.8.13 'cmake' Invocation

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

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

See also:

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

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

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

The `'cmake'` script produces an output like this:

```
export LIBSUFFIX_4_CMAKE="-DLIB_SUFFIX=64"
export INSTALL_BASEDIR=/home/user/dev/deliveries
cmake -DCMAKE_INSTALL_PREFIX=${INSTALL_BASEDIR}/simcrs-0.1.0 \
-DWITH_STDAIR_PREFIX=${INSTALL_BASEDIR}/stdair-stable \
-DWITH_TRADEMGEN_PREFIX=${INSTALL_BASEDIR}/trademgen-stable \
-DWITH_TRAVELCCM_PREFIX=${INSTALL_BASEDIR}/travelccm-stable \
-DWITH_AIRSCHED_PREFIX=${INSTALL_BASEDIR}/airsched-stable \
-DWITH_AIRRRAC_PREFIX=${INSTALL_BASEDIR}/airrac-stable \
-DWITH_RMOL_PREFIX=${INSTALL_BASEDIR}/rmol-stable \
-DWITH_AIRINV_PREFIX=${INSTALL_BASEDIR}/airinv-stable \
-DWITH_SIMFQT_PREFIX=${INSTALL_BASEDIR}/simfqt-stable \
-DCMAKE_BUILD_TYPE: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: 7a1519ef5b14232c47felb1d46db4ae9e65e696d trunk
-- Requires Boost-1.41
-- Boost version: 1.47.0
-- Found the following Boost libraries:
--   regex
--   program_options
--   date_time
--   iostreams
--   serialization
--   filesystem
--   unit_test_framework
--   python
-- Found Boost version: 1.47.0
-- Found BoostWrapper: /usr/include (found suitable version "1.47.0", required is "1.41")
-- Requires Readline without specifying any version
-- Found Readline: /usr/include (found version "6.2")
-- Found Readline version: 6.2
-- Requires MySQL without specifying any version
-- Using mysql-config: /usr/bin/mysql_config
-- Found MySQL: /usr/lib64/mysql/libmysqlclient.so (found version "5.5.18")
-- Found MySQL version: 5.5.18
-- Requires SOCI-3.0
-- SOCI headers are not buried
-- Found SOCI: /usr/lib64/libsoci_core.so (found suitable version "3.1.0", required is "3.0")
-- Found SOCIMySQL: /usr/lib64/libsoci_mysql.so (found suitable version "3.1.0", required is "3.0")
-- Found SOCI with MySQL back-end support version: 3.1.0
-- Requires StdAir-0.43
-- Found StdAir version: 0.44.3
-- Requires AirSched-0.1
-- Found AirSched version: 0.1.3
```

```
-- Requires AirRAC-0.2
-- Found AirRAC version: 0.2.2
-- Requires RMOL-0.25
-- Found RMOL version: 0.25.2
-- Requires AirInv-0.1
-- Found AirInv version: 0.1.2
-- Requires SimFQT-0.1
-- Found SimFQT version: 0.1.2
-- Requires Doxygen without specifying any version
-- Found Doxygen: /usr/bin/doxygen
-- Found DoxygenWrapper: /usr/bin/doxygen (found version "1.7.5")
-- Found Doxygen version: 1.7.5
-- Had to set the linker language for 'simcrslib' to CXX
-- Test 'CRSTestSuite' to be built with 'CRSTestSuite.cpp'
--
-- =====
--
-- ---      Project Information      ---
--
-- PROJECT_NAME ..... : simcrs
-- PACKAGE_PRETTY_NAME ..... : SimCRS
-- PACKAGE ..... : simcrs
-- PACKAGE_NAME ..... : SIMCRS
-- PACKAGE_BRIEF ..... : C++ Simulated Travel-Oriented Distribution System Library
-- PACKAGE_VERSION ..... : 0.5.0
-- GENERIC_LIB_VERSION ..... : 0.5.0
-- GENERIC_LIB_SOVERSION ..... : 0.5
--
-- ---      Build Configuration      ---
--
-- Modules to build ..... : simcrs
-- Libraries to build/install ..... : simcrslib
-- Binaries to build/install ..... : simcrs
-- Modules to test ..... : simcrs
-- Binaries to test ..... : CRSTestSuitetst
--
-- * Module ..... : simcrs
--   + Layers to build ..... : ;basic;bom;factory;command;service
--   + Dependencies on other layers .. :
--   + Libraries to build/install .... : simcrslib
--   + Executables to build/install .. : simcrs
--   + Tests to perform ..... : CRSTestSuitetst
--
-- BUILD_SHARED_LIBS ..... : ON
-- CMAKE_BUILD_TYPE ..... : Debug
-- * CMAKE_C_FLAGS ..... :
-- * CMAKE_CXX_FLAGS ..... : -Wall -Werror -DBOOST_VERSION=104700
-- * BUILD_FLAGS ..... :
-- * COMPILE_FLAGS ..... :
-- CMAKE_MODULE_PATH ..... : /home/user/dev/sim/simcrs/simcrsgithub/config/
-- CMAKE_INSTALL_PREFIX ..... : /home/user/dev/deliveries/simcrs-0.5.0
--
-- * Doxygen:
--   - DOXYGEN_VERSION ..... : 1.7.5
--   - DOXYGEN_EXECUTABLE ..... : /usr/bin/doxygen
--   - DOXYGEN_DOT_EXECUTABLE ..... : /usr/bin/dot
--   - DOXYGEN_DOT_PATH ..... : /usr/bin
--
-- ---      Installation Configuration      ---
--
-- INSTALL_LIB_DIR ..... : /home/user/dev/deliveries/simcrs-0.5.0/lib64
-- INSTALL_BIN_DIR ..... : /home/user/dev/deliveries/simcrs-0.5.0/bin
-- CMAKE_INSTALL_RPATH ..... : /home/user/dev/deliveries/simcrs-0.5.0/lib64
-- CMAKE_INSTALL_RPATH_USE_LINK_PATH . : ON
-- INSTALL_INCLUDE_DIR ..... : /home/user/dev/deliveries/simcrs-0.5.0/include
```



```

-- INSTALL_DATA_DIR ..... : /home/user/dev/deliveries/simcrs-0.5.0/share
-- INSTALL_SAMPLE_DIR ..... : /home/user/dev/deliveries/simcrs-0.5.0/share/simcrs/samples
-- INSTALL_DOC ..... : ON
--
-- -----
-- ---      Packaging Configuration      ---
-- -----
-- CPACK_PACKAGE_CONTACT ..... : Denis Arnaud <denis_arnaud - at - users dot sourceforge dot net>
-- CPACK_PACKAGE_VENDOR ..... : Denis Arnaud
-- CPACK_PACKAGE_VERSION ..... : 0.5.0
-- CPACK_PACKAGE_DESCRIPTION_FILE .... : /home/user/dev/sim/simcrs/simcrsgithub/README
-- CPACK_RESOURCE_FILE_LICENSE ..... : /home/user/dev/sim/simcrs/simcrsgithub/COPYING
-- CPACK_GENERATOR ..... : TBZ2
-- CPACK_DEBIAN_PACKAGE_DEPENDS ..... :
-- CPACK_SOURCE_GENERATOR ..... : TBZ2;TGZ
-- CPACK_SOURCE_PACKAGE_FILE_NAME .... : simcrs-0.5.0
--
-- -----
-- ---      External libraries      ---
-- -----
--
-- * Boost:
--   - Boost_VERSION ..... : 104700
--   - Boost_LIB_VERSION ..... : 1_47
--   - Boost_HUMAN_VERSION ..... : 1.47.0
--   - Boost_INCLUDE_DIRS ..... : /usr/include
--   - Boost required components ..... : regex;program_options;date_time;iostreams;serialization;filesystem
--   - Boost required libraries ..... : /usr/lib64/libboost_regex-mt.so;/usr/lib64/libboost_iostreams-mt.so
--
-- * Readline:
--   - READLINE_VERSION ..... : 6.2
--   - READLINE_INCLUDE_DIR ..... : /usr/include
--   - READLINE_LIBRARY ..... : /usr/lib64/libreadline.so
--
-- * MySQL:
--   - MYSQL_VERSION ..... : 5.5.18
--   - MYSQL_INCLUDE_DIR ..... : /usr/include/mysql
--   - MYSQL_LIBRARIES ..... : /usr/lib64/mysql/libmysqlclient.so
--
-- * SOCI:
--   - SOCI_VERSION ..... : 300100
--   - SOCI_LIB_VERSION ..... : 3_1_0
--   - SOCI_HUMAN_VERSION ..... : 3.1.0
--   - SOCI_INCLUDE_DIR ..... : /usr/include/soci
--   - SOCIMYSQL_INCLUDE_DIR ..... : /usr/include/soci/mysql
--   - SOCI_LIBRARIES ..... : /usr/lib64/libsoci_core.so
--   - SOCIMYSQL_LIBRARIES ..... : /usr/lib64/libsoci_mysql.so
--
-- * StdAir:
--   - STDAIR_VERSION ..... : 0.44.3
--   - STDAIR_BINARY_DIRS ..... : /home/user/dev/deliveries/stdair-0.44.3/bin
--   - STDAIR_EXECUTABLES ..... : stdair
--   - STDAIR_LIBRARY_DIRS ..... : /home/user/dev/deliveries/stdair-0.44.3/lib64
--   - STDAIR_LIBRARIES ..... : stdairlib;stdairuicllib
--   - STDAIR_INCLUDE_DIRS ..... : /home/user/dev/deliveries/stdair-0.44.3/include
--   - STDAIR_SAMPLE_DIR ..... : /home/user/dev/deliveries/stdair-0.44.3/share/stdair/samples
--
-- * AirSched:
--   - AIRSCHED_VERSION ..... : 0.1.3
--   - AIRSCHED_BINARY_DIRS ..... : /home/user/dev/deliveries/airsched-0.1.3/bin
--   - AIRSCHED_EXECUTABLES ..... : airsched
--   - AIRSCHED_LIBRARY_DIRS ..... : /home/user/dev/deliveries/airsched-0.1.3/lib64
--   - AIRSCHED_LIBRARIES ..... : airschedlib
--   - AIRSCHED_INCLUDE_DIRS ..... : /home/user/dev/deliveries/airsched-0.1.3/include
--
-- * AirRAC:
--   - AIRRAC_VERSION ..... : 0.2.2

```

```
-- - AIRRAC_BINARY_DIRS ..... : /home/user/dev/deliveries/airrac-0.2.2/bin
-- - AIRRAC_EXECUTABLES ..... : airrac
-- - AIRRAC_LIBRARY_DIRS ..... : /home/user/dev/deliveries/airrac-0.2.2/lib64
-- - AIRRAC_LIBRARIES ..... : airraclib
-- - AIRRAC_INCLUDE_DIRS ..... : /home/user/dev/deliveries/airrac-0.2.2/include
--
-- * RMOL:
-- - RMOL_VERSION ..... : 0.25.2
-- - RMOL_BINARY_DIRS ..... : /home/user/dev/deliveries/rmol-0.25.2/bin
-- - RMOL_EXECUTABLES ..... : rmol
-- - RMOL_LIBRARY_DIRS ..... : /home/user/dev/deliveries/rmol-0.25.2/lib
-- - RMOL_LIBRARIES ..... : rmolllib
-- - RMOL_INCLUDE_DIRS ..... : /home/user/dev/deliveries/rmol-0.25.2/include
--
-- * AirInv:
-- - AIRINV_VERSION ..... : 0.1.2
-- - AIRINV_BINARY_DIRS ..... : /home/user/dev/deliveries/airinv-0.1.2/bin
-- - AIRINV_EXECUTABLES ..... : airinv;airinv_parseInventory
-- - AIRINV_LIBRARY_DIRS ..... : /home/user/dev/deliveries/airinv-0.1.2/lib
-- - AIRINV_LIBRARIES ..... : airinvlib
-- - AIRINV_INCLUDE_DIRS ..... : /home/user/dev/deliveries/airinv-0.1.2/include
--
-- * SimFQT:
-- - SIMFQT_VERSION ..... : 0.1.2
-- - SIMFQT_BINARY_DIRS ..... : /home/user/dev/deliveries/simfqt-0.1.2/bin
-- - SIMFQT_EXECUTABLES ..... : simfqt;simfqt_parseFareRules
-- - SIMFQT_LIBRARY_DIRS ..... : /home/user/dev/deliveries/simfqt-0.1.2/lib64
-- - SIMFQT_LIBRARIES ..... : simfqtlib
-- - SIMFQT_INCLUDE_DIRS ..... : /home/user/dev/deliveries/simfqt-0.1.2/include
--
-- Change a value with: cmake -D<Variable>=<Value>
-- =====
--
-- Configuring done
-- Generating done
-- Build files have been written to: /home/user/dev/sim/simcrs/simcrsgithub/build
```

It is recommended that you check if your library has been compiled and linked properly and works as expected. To do so, you should execute the testing process 'make check'. As a result, you should obtain a similar report:

```
[ 0%] Built target hdr_cfg_simcrs
[ 90%] Built target simcrslib
[100%] Built target CRSTestSuitetst
Scanning dependencies of target check_simcrstst
Test project /home/user/dev/sim/simcrs/simcrsgithub/build/test/simcrs
  Start 1: CRSTestSuitetst
1/1 Test #1: CRSTestSuitetst ..... Passed    0.15 sec

100% tests passed, 0 tests failed out of 1

Total Test time (real) = 0.33 sec
[100%] Built target check_simcrstst
Scanning dependencies of target check
[100%] Built target check
```

Check if all the executed tests PASSED. If not, please contact us by filling a [bug-report](#).

Finally, you should install the compiled and linked library, include files and (optionally) HTML and PDF documentation by typing:

```
make install
```

Depending on the PREFIX settings during configuration, you might need the root (administrator) access to perform this step.

Eventually, you might invoke the following command

```
make clean
```

to remove all files created during compilation process, or even

```
cd ~/dev/sim/simcrsgit
rm -rf build && mkdir build
cd build
```

to remove everything.

12.9 Linking with SimCRS

12.9.1 Table of Contents

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

12.9.2 Introduction

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

12.9.3 Dependencies

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

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

12.9.3.2 Other Simulation-Related Components SimCRS, as shown on the diagram below, depends on

- [AirSched](#)
- [SimFQT](#)
- [AirRAC](#)

- RMOL
- AirInv
- AvlCal
- SimLFS

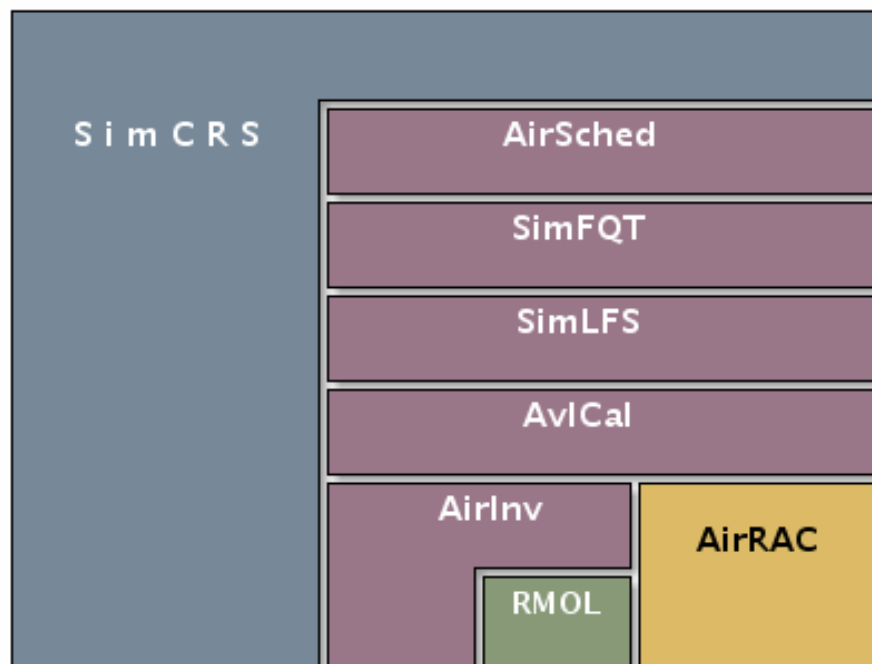


Figure 1: SimCRS Dependencies

12.9.4 Using the pkg-config command

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

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

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

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

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

12.9.5 Using the `simcrs-config` script

SimCRS provides a shell script called `'simcrs-config'`, which is installed by default in `'$prefix/bin'` (`'/usr/local/bin'`) directory. It can be used to simplify compilation and linking of SimCRS based programs. The usage of this script is quite similar to the usage of the `'pkg-config'` command.

Assuming that you need to compile the program `'my_prog.cpp'` you can now do that with the following command:

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

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

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

If the `'simcrs-config'` command is not found by your shell, you should add its location `'$prefix/bin'` to the `PATH` environment variable, e.g.:

```
export PATH=/usr/local/bin:$PATH
```

12.9.6 M4 macro for the GNU Autotools

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

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

12.9.7 Using SimCRS with dynamic linking

When using static linking some of the library routines in SimCRS are copied into your executable program. This can lead to unnecessary large executables. To avoid having too large executable files you may use dynamic linking instead. Dynamic linking means that the actual linking is performed when the program is executed. This requires that the system is able to locate the shared SimCRS library file during your program execution. If you install the SimCRS library using a non-standard prefix, the `'LD_LIBRARY_PATH'` environment variable might be used to inform the linker of the dynamic library location, e.g.:

```
export LD_LIBRARY_PATH=<SimCRS installation prefix>/lib:$LD_LIBRARY_PATH
```

12.10 Test Rules

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

12.10.1 The Test Source Files

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

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

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

12.10.2 The Reference File

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

12.10.3 Testing SimCRS Library

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

```
% make check
```

after successful compilation of the SimCRS library.

12.11 Users Guide

12.11.1 Table of Contents

- [Introduction](#)
- [Get Started](#)
 - [Get the SimCRS library](#)
 - [Build the SimCRS project](#)
 - [Build and Run the Tests](#)
 - [Install the SimCRS Project \(Binaries, Documentation\)](#)
- [Input file of SimCRS Project](#)
- [The schedule BOM Tree](#)
 - [Build of the schedule BOM tree](#)
 - [Display of the schedule BOM tree](#)
- [Exploring the Predefined BOM Tree](#)
 - [Airline Network BOM Tree](#)
 - [Airline Schedule BOM Tree](#)

- [Extending the BOM Tree](#)
- [The travel solution calculation procedure](#)

12.11.2 Introduction

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

12.11.3 Get Started

12.11.3.1 Get the SimCRS library Clone locally the full [Git project](#):

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

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

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

12.11.3.3 Build and Run the Tests After building the SimCRS project, the following commands run the tests:

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

As a result, you should obtain a similar report:

```
[ 0%] Built target hdr_cfg_simcrs
[ 96%] Built target simcrslib
[100%] Built target AirlineScheduleTestSuitetst
Scanning dependencies of target check_simcrstst
Test project /home/dan/dev/sim/simcrs/simcrsgithub/build/test/simcrs
  Start 1: AirlineScheduleTestSuitetst
1/1 Test #1: AirlineScheduleTestSuitetst ..... Passed    0.15 sec

100% tests passed, 0 tests failed out of 1

Total Test time (real) = 0.40 sec
[100%] Built target check_simcrstst
Scanning dependencies of target check
[100%] Built target check
```

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

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

You can check that the executables and other required files have been copied into the given final directory:

```
cd ~/dev/deliveries/simcrs-0.5.0
```

To generate the SimCRS project documentation, the commands are:

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

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

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

12.11.4 Input file of SimCRS Project

The schedule input file structure should look like the following sample:

Each line, beyond the header, represents a schedule entry, i.e., the specification of a given flight-period (see `SIMCRS::FlightPeriodStruct`). The fields are as follows:

- Flights section
 - AirlineCode (e.g., BA)
 - FlightNumber (e.g., 9)
 - Start of the flight departure period (e.g., 2007-04-20)
 - End of the flight departure period (e.g., 2007-06-30)
 - Day-Of-the-Week for the flight departure period (DOW) (e.g., 0000011)
 - Leg section
 - Segment section
- Leg section
 - BoardPoint (e.g., LHR)
 - OffPoint (e.g., BKK)
 - BoardTime (e.g., 22:00)
 - ArrivalTime (e.g., 15:15)
 - ArrivalDateOffset (e.g., +1)
 - ElapsedTime (e.g., 11:15)

- Leg-cabin section
- Leg-cabin section
 - Cabin code (e.g., F, J, W or Y)
 - Capacity (e.g., respectively 5, 12, 20 or 300)
- Segment section
 - Specificity flag:
 - * 0 means that all the segments behave the same way, i.e., have got the same dressing (distribution and order of the booking classes per cabin)
 - * 1 means that each segment behave differently. The full specification of each of those segments must therefore be given.
 - Segment-cabin section
 - Fare family section
- Segment-cabin section
 - Cabin code (e.g., F, J, W or Y)
 - List of (one-letter-code) booking classes for the cabin (e.g, respectively FA, JC DI, WT or YBHKMLSQ)
- Fare family section
 - Fare family code (e.g., 1)
 - List of (one-letter-code) booking classes for the fare family (e.g, respectively FA, JC DI, WT or YBHKMLSQ)

Some fare input examples (including the example above named `schedule03.csv`) are given in the StdAir project.

12.11.5 The schedule BOM Tree

The schedule-related Business Object Model (BOM) tree is a structure allowing to store all the `SIMCRS::FlightPeriodStruct` objects of the simulation. That is why parsing an input file, containing the specification for all the flight-periods, is more convenient (

See also:

the previous section [Input file of SimCRS Project](#)).

As it may be time consuming, and it for sure requires some know-how, to first build such a schedule input file, a small sample BOM tree is provided by default when needed.

12.11.5.1 Build of the schedule BOM tree First, a BOM root object (i.e., a root for all the classes in the project) is instantiated by the `stdair::STDAIR_ServiceContext` context object, when the `stdair::STDAIR_Service` is itself instantiated (during the instantiation of the `SIMCRS::SIMCRS_Service` object).

The corresponding type (class) `stdair::BomRoot` is defined in the StdAir library.

Then, the BOM root can be either constructed thanks to the `SIMCRS::SIMCRS_Service::buildSampleBom()` method:

```
void buildSampleBom ();
```

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

```
void parseAndLoad (const stdair::ScheduleFilePath&,
```

12.11.5.2 Display of the schedule BOM tree

Note:

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

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

When the default BOM tree is used (-b/-builtin option of the main program `simcrs.cpp`), the schedule BOM tree display (for now, corresponding to `schedule01.csv` parsed by `SIMCRS::parseInventory`) should look like:

```
=====
BomRoot:  -- ROOT  --
=====
+++++
Inventory: SQ
+++++
*****
FlightDate: SQ11, 2010-Jan-15
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-15, SIN-BKK, 2010-Jan-15, 08:20:00, 2010-Jan-15, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-15, SIN-BKK 2010-Jan-15, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 2, 298, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-15, SIN-BKK 2010-Jan-15, Y, 1, 0, 0, 0, 2, 298, 0,
SQ11 2010-Jan-15, SIN-BKK 2010-Jan-15, Y, 2, 0, 0, 0, 2, 298, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-15, SIN-BKK 2010-Jan-15, Y, 1, Y, 300 (0), 0, 0, 0, 2, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Jan-15, SIN-BKK 2010-Jan-15, Y, 2, M, 300 (0), 0, 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
```

```
*****
*****
FlightDate: SQ11, 2010-Jan-16
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-16, SIN-BKK, 2010-Jan-16, 08:20:00, 2010-Jan-16, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-16, SIN-BKK 2010-Jan-16, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 300, 9, 1.83244e-319, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-16, SIN-BKK 2010-Jan-16, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-16, SIN-BKK 2010-Jan-16, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-16, SIN-BKK 2010-Jan-16, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Jan-16, SIN-BKK 2010-Jan-16, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Jan-17
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-17, SIN-BKK, 2010-Jan-17, 08:20:00, 2010-Jan-17, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-17, SIN-BKK 2010-Jan-17, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 1.58896e-319, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-17, SIN-BKK 2010-Jan-17, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-17, SIN-BKK 2010-Jan-17, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-17, SIN-BKK 2010-Jan-17, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
```

```
SQL1 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: SQL1, 2010-Jan-18
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 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,
SQL1 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,
SQL1 2010-Jan-18, SIN-BKK 2010-Jan-18, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 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,
SQL1 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,
SQL1 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: SQL1, 2010-Jan-19
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 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,
SQL1 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,
SQL1 2010-Jan-19, SIN-BKK 2010-Jan-19, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 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,
```

```
SQL1 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,
SQL1 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: SQL1, 2010-Jan-20
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 2010-Jan-20, SIN-BKK, 2010-Jan-20, 08:20:00, 2010-Jan-20, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL1 2010-Jan-20, SIN-BKK 2010-Jan-20, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL1 2010-Jan-20, SIN-BKK 2010-Jan-20, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 2010-Jan-20, SIN-BKK 2010-Jan-20, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQL1 2010-Jan-20, SIN-BKK 2010-Jan-20, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQL1 2010-Jan-20, SIN-BKK 2010-Jan-20, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL1, 2010-Jan-21
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 2010-Jan-21, SIN-BKK, 2010-Jan-21, 08:20:00, 2010-Jan-21, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL1 2010-Jan-21, SIN-BKK 2010-Jan-21, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL1 2010-Jan-21, SIN-BKK 2010-Jan-21, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 2010-Jan-21, SIN-BKK 2010-Jan-21, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
```

```
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-21, SIN-BKK 2010-Jan-21, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Jan-21, SIN-BKK 2010-Jan-21, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Jan-22
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-22, SIN-BKK, 2010-Jan-22, 08:20:00, 2010-Jan-22, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-22, SIN-BKK 2010-Jan-22, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-22, SIN-BKK 2010-Jan-22, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-22, SIN-BKK 2010-Jan-22, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-22, SIN-BKK 2010-Jan-22, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Jan-22, SIN-BKK 2010-Jan-22, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Jan-23
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-23, SIN-BKK, 2010-Jan-23, 08:20:00, 2010-Jan-23, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-23, SIN-BKK 2010-Jan-23, Y, 300, 300, 0, 0, 0, 0, 0, 0, 6.64029e-319, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-23, SIN-BKK 2010-Jan-23, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-23, SIN-BKK 2010-Jan-23, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
```

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-23, SIN-BKK 2010-Jan-23, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Jan-23, SIN-BKK 2010-Jan-23, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ11, 2010-Jan-24

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-24, SIN-BKK, 2010-Jan-24, 08:20:00, 2010-Jan-24, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-24, SIN-BKK 2010-Jan-24, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,

SQ11 2010-Jan-24, SIN-BKK 2010-Jan-24, Y, 1, 0, 0, 0, 0, 300, 0,

SQ11 2010-Jan-24, SIN-BKK 2010-Jan-24, Y, 2, 0, 0, 0, 0, 300, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,

SQ11 2010-Jan-24, SIN-BKK 2010-Jan-24, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

SQ11 2010-Jan-24, SIN-BKK 2010-Jan-24, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ11, 2010-Jan-25

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-25, SIN-BKK, 2010-Jan-25, 08:20:00, 2010-Jan-25, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,

SQ11 2010-Jan-25, SIN-BKK 2010-Jan-25, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,

SQ11 2010-Jan-25, SIN-BKK 2010-Jan-25, Y, 1, 0, 0, 0, 0, 300, 0,

SQ11 2010-Jan-25, SIN-BKK 2010-Jan-25, Y, 2, 0, 0, 0, 0, 300, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-25, SIN-BKK 2010-Jan-25, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Jan-25, SIN-BKK 2010-Jan-25, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ11, 2010-Jan-26

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-26, SIN-BKK, 2010-Jan-26, 08:20:00, 2010-Jan-26, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-26, SIN-BKK 2010-Jan-26, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,

SQ11 2010-Jan-26, SIN-BKK 2010-Jan-26, Y, 1, 0, 0, 0, 0, 0, 300, 0,

SQ11 2010-Jan-26, SIN-BKK 2010-Jan-26, Y, 2, 0, 0, 0, 0, 0, 300, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-26, SIN-BKK 2010-Jan-26, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Jan-26, SIN-BKK 2010-Jan-26, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ11, 2010-Jan-27

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-27, SIN-BKK, 2010-Jan-27, 08:20:00, 2010-Jan-27, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-27, SIN-BKK 2010-Jan-27, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,

SQ11 2010-Jan-27, SIN-BKK 2010-Jan-27, Y, 1, 0, 0, 0, 0, 0, 300, 0,

SQ11 2010-Jan-27, SIN-BKK 2010-Jan-27, Y, 2, 0, 0, 0, 0, 0, 300, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-27, SIN-BKK 2010-Jan-27, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Jan-27, SIN-BKK 2010-Jan-27, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ11, 2010-Jan-28

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-28, SIN-BKK, 2010-Jan-28, 08:20:00, 2010-Jan-28, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-28, SIN-BKK 2010-Jan-28, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-28, SIN-BKK 2010-Jan-28, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-28, SIN-BKK 2010-Jan-28, Y, 2, 0, 0, 0, 0, 300, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-28, SIN-BKK 2010-Jan-28, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Jan-28, SIN-BKK 2010-Jan-28, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ11, 2010-Jan-29

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-29, SIN-BKK, 2010-Jan-29, 08:20:00, 2010-Jan-29, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-29, SIN-BKK 2010-Jan-29, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-29, SIN-BKK 2010-Jan-29, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-29, SIN-BKK 2010-Jan-29, Y, 2, 0, 0, 0, 0, 300, 0,

```
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-29, SIN-BKK 2010-Jan-29, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Jan-29, SIN-BKK 2010-Jan-29, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Jan-30
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-30, SIN-BKK, 2010-Jan-30, 08:20:00, 2010-Jan-30, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-30, SIN-BKK 2010-Jan-30, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-30, SIN-BKK 2010-Jan-30, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Jan-30, SIN-BKK 2010-Jan-30, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Jan-30, SIN-BKK 2010-Jan-30, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Jan-30, SIN-BKK 2010-Jan-30, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Jan-31
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Jan-31, SIN-BKK, 2010-Jan-31, 08:20:00, 2010-Jan-31, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Jan-31, SIN-BKK 2010-Jan-31, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Jan-31, SIN-BKK 2010-Jan-31, Y, 1, 0, 0, 0, 0, 300, 0,
```

```
SQL1 2010-Jan-31, SIN-BKK 2010-Jan-31, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQL1 2010-Jan-31, SIN-BKK 2010-Jan-31, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQL1 2010-Jan-31, SIN-BKK 2010-Jan-31, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL1, 2010-Feb-01
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 2010-Feb-01, SIN-BKK, 2010-Feb-01, 08:20:00, 2010-Feb-01, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL1 2010-Feb-01, SIN-BKK 2010-Feb-01, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL1 2010-Feb-01, SIN-BKK 2010-Feb-01, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 2010-Feb-01, SIN-BKK 2010-Feb-01, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQL1 2010-Feb-01, SIN-BKK 2010-Feb-01, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQL1 2010-Feb-01, SIN-BKK 2010-Feb-01, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL1, 2010-Feb-02
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 2010-Feb-02, SIN-BKK, 2010-Feb-02, 08:20:00, 2010-Feb-02, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL1 2010-Feb-02, SIN-BKK 2010-Feb-02, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
```

```
SQL1 2010-Feb-02, SIN-BKK 2010-Feb-02, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 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,
SQL1 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,
SQL1 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: SQL1, 2010-Feb-03
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 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,
SQL1 2010-Feb-03, SIN-BKK 2010-Feb-03, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL1 2010-Feb-03, SIN-BKK 2010-Feb-03, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 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,
SQL1 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,
SQL1 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: SQL1, 2010-Feb-04
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 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,
SQL1 2010-Feb-04, SIN-BKK 2010-Feb-04, 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-04, SIN-BKK 2010-Feb-04, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-04, SIN-BKK 2010-Feb-04, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-04, SIN-BKK 2010-Feb-04, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-04, SIN-BKK 2010-Feb-04, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-05
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-05, SIN-BKK, 2010-Feb-05, 08:20:00, 2010-Feb-05, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-05, SIN-BKK 2010-Feb-05, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-05, SIN-BKK 2010-Feb-05, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-05, SIN-BKK 2010-Feb-05, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-05, SIN-BKK 2010-Feb-05, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-05, SIN-BKK 2010-Feb-05, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-06
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-06, SIN-BKK, 2010-Feb-06, 08:20:00, 2010-Feb-06, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-06, SIN-BKK 2010-Feb-06, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
```

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

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-08, SIN-BKK 2010-Feb-08, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-08, SIN-BKK 2010-Feb-08, Y, 2, 0, 0, 0, 0, 300, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-08, SIN-BKK 2010-Feb-08, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-08, SIN-BKK 2010-Feb-08, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ11, 2010-Feb-09

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-09, SIN-BKK, 2010-Feb-09, 08:20:00, 2010-Feb-09, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-09, SIN-BKK 2010-Feb-09, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-09, SIN-BKK 2010-Feb-09, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-09, SIN-BKK 2010-Feb-09, Y, 2, 0, 0, 0, 0, 300, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-09, SIN-BKK 2010-Feb-09, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-09, SIN-BKK 2010-Feb-09, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ11, 2010-Feb-10

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-10, SIN-BKK, 2010-Feb-10, 08:20:00, 2010-Feb-10, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-10, SIN-BKK 2010-Feb-10, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-10, SIN-BKK 2010-Feb-10, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-10, SIN-BKK 2010-Feb-10, Y, 2, 0, 0, 0, 0, 300, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-10, SIN-BKK 2010-Feb-10, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-10, SIN-BKK 2010-Feb-10, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ11, 2010-Feb-11

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-11, SIN-BKK, 2010-Feb-11, 08:20:00, 2010-Feb-11, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-11, SIN-BKK 2010-Feb-11, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-11, SIN-BKK 2010-Feb-11, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-11, SIN-BKK 2010-Feb-11, Y, 2, 0, 0, 0, 0, 300, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-11, SIN-BKK 2010-Feb-11, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-11, SIN-BKK 2010-Feb-11, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ11, 2010-Feb-12

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-12, SIN-BKK, 2010-Feb-12, 08:20:00, 2010-Feb-12, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-12, SIN-BKK 2010-Feb-12, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,


```
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-12, SIN-BKK 2010-Feb-12, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-12, SIN-BKK 2010-Feb-12, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-12, SIN-BKK 2010-Feb-12, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0,
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,
*****
*****
FlightDate: SQ11, 2010-Feb-13
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-13, SIN-BKK, 2010-Feb-13, 08:20:00, 2010-Feb-13, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-13, SIN-BKK 2010-Feb-13, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-13, SIN-BKK 2010-Feb-13, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-13, SIN-BKK 2010-Feb-13, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-13, SIN-BKK 2010-Feb-13, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0,
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,
*****
*****
FlightDate: SQ11, 2010-Feb-14
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-14, SIN-BKK, 2010-Feb-14, 08:20:00, 2010-Feb-14, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-14, SIN-BKK 2010-Feb-14, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0,
*****
*****
Buckets:
-----
```

```
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-14, SIN-BKK 2010-Feb-14, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-14, SIN-BKK 2010-Feb-14, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-14, SIN-BKK 2010-Feb-14, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-14, SIN-BKK 2010-Feb-14, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-15
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-15, SIN-BKK, 2010-Feb-15, 08:20:00, 2010-Feb-15, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-15, SIN-BKK 2010-Feb-15, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-15, SIN-BKK 2010-Feb-15, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-15, SIN-BKK 2010-Feb-15, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-15, SIN-BKK 2010-Feb-15, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-15, SIN-BKK 2010-Feb-15, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-16
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-16, SIN-BKK, 2010-Feb-16, 08:20:00, 2010-Feb-16, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-16, SIN-BKK 2010-Feb-16, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
```

```
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-16, SIN-BKK 2010-Feb-16, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-16, SIN-BKK 2010-Feb-16, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-16, SIN-BKK 2010-Feb-16, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-16, SIN-BKK 2010-Feb-16, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-17
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-17, SIN-BKK, 2010-Feb-17, 08:20:00, 2010-Feb-17, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-17, SIN-BKK 2010-Feb-17, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-17, SIN-BKK 2010-Feb-17, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-17, SIN-BKK 2010-Feb-17, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-17, SIN-BKK 2010-Feb-17, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-17, SIN-BKK 2010-Feb-17, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ11, 2010-Feb-18
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-18, SIN-BKK, 2010-Feb-18, 08:20:00, 2010-Feb-18, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-18, SIN-BKK 2010-Feb-18, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
```

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-18, SIN-BKK 2010-Feb-18, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-18, SIN-BKK 2010-Feb-18, Y, 2, 0, 0, 0, 0, 300, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-18, SIN-BKK 2010-Feb-18, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-18, SIN-BKK 2010-Feb-18, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ11, 2010-Feb-19

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-19, SIN-BKK, 2010-Feb-19, 08:20:00, 2010-Feb-19, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-19, SIN-BKK 2010-Feb-19, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-19, SIN-BKK 2010-Feb-19, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-19, SIN-BKK 2010-Feb-19, Y, 2, 0, 0, 0, 0, 300, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-19, SIN-BKK 2010-Feb-19, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-19, SIN-BKK 2010-Feb-19, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ11, 2010-Feb-20

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-20, SIN-BKK, 2010-Feb-20, 08:20:00, 2010-Feb-20, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-20, SIN-BKK 2010-Feb-20, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-20, SIN-BKK 2010-Feb-20, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-20, SIN-BKK 2010-Feb-20, Y, 2, 0, 0, 0, 0, 300, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-20, SIN-BKK 2010-Feb-20, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-20, SIN-BKK 2010-Feb-20, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ11, 2010-Feb-21

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-21, SIN-BKK, 2010-Feb-21, 08:20:00, 2010-Feb-21, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-21, SIN-BKK 2010-Feb-21, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-21, SIN-BKK 2010-Feb-21, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-21, SIN-BKK 2010-Feb-21, Y, 2, 0, 0, 0, 0, 300, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-21, SIN-BKK 2010-Feb-21, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-21, SIN-BKK 2010-Feb-21, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ11, 2010-Feb-22

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-22, SIN-BKK, 2010-Feb-22, 08:20:00, 2010-Feb-22, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-22, SIN-BKK 2010-Feb-22, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,

```
*****
*****
Buckets:
-----
```

```
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
```

```
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-22, SIN-BKK 2010-Feb-22, Y, 1, 0, 0, 0, 0, 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,
SQ11 2010-Feb-22, SIN-BKK 2010-Feb-22, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-22, SIN-BKK 2010-Feb-22, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
```

```
*****
*****
FlightDate: SQ11, 2010-Feb-23
*****
*****
```

```
Leg-Dates:
-----
```

```
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ11 2010-Feb-23, SIN-BKK, 2010-Feb-23, 08:20:00, 2010-Feb-23, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
```

```
LegCabins:
-----
```

```
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-23, SIN-BKK 2010-Feb-23, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
```

```
*****
*****
Buckets:
-----
```

```
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
```

```
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-23, SIN-BKK 2010-Feb-23, Y, 1, 0, 0, 0, 0, 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,
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,
```

```
SQL1 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,
SQL1 2010-Feb-24, SIN-BKK 2010-Feb-24, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 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,
SQL1 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,
SQL1 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: SQL1, 2010-Feb-25
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 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,
SQL1 2010-Feb-25, SIN-BKK 2010-Feb-25, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL1 2010-Feb-25, SIN-BKK 2010-Feb-25, Y, 1, 0, 0, 0, 0, 300, 0,
SQL1 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,
SQL1 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,
SQL1 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: SQL1, 2010-Feb-26
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL1 2010-Feb-26, SIN-BKK, 2010-Feb-26, 08:20:00, 2010-Feb-26, 11:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
```

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



```
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ11 2010-Feb-28, SIN-BKK 2010-Feb-28, Y, 300, 300, 0, 0, 0, 0, 0, 0, 0, 0, 300, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ11 2010-Feb-28, SIN-BKK 2010-Feb-28, Y, 1, 0, 0, 0, 0, 300, 0,
SQ11 2010-Feb-28, SIN-BKK 2010-Feb-28, Y, 2, 0, 0, 0, 0, 300, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ11 2010-Feb-28, SIN-BKK 2010-Feb-28, Y, 1, Y, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ11 2010-Feb-28, SIN-BKK 2010-Feb-28, Y, 2, M, 300 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Jan-15
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Jan-15, SIN-HND, 2010-Jan-15, 09:20:00, 2010-Jan-15, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Jan-15, SIN-HND 2010-Jan-15, Y, 200, 200, 2.082e+121, 5.53287e-48, 5.20268e-90, 0, 1.31346e-47,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Jan-15, SIN-HND 2010-Jan-15, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Jan-15, SIN-HND 2010-Jan-15, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Jan-15, SIN-HND 2010-Jan-15, Y, 1, Y13856, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Jan-15, SIN-HND 2010-Jan-15, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Jan-16
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Jan-16, SIN-HND, 2010-Jan-16, 09:20:00, 2010-Jan-16, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
```

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Jan-16, SIN-HND 2010-Jan-16, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 2.63638e-319, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Jan-16, SIN-HND 2010-Jan-16, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Jan-16, SIN-HND 2010-Jan-16, Y, 2, 0, 0, 0, 0, 200, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Jan-16, SIN-HND 2010-Jan-16, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Jan-16, SIN-HND 2010-Jan-16, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ12, 2010-Jan-17

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Jan-17, SIN-HND, 2010-Jan-17, 09:20:00, 2010-Jan-17, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Jan-17, SIN-HND 2010-Jan-17, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 2.39291e-319, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Jan-17, SIN-HND 2010-Jan-17, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Jan-17, SIN-HND 2010-Jan-17, Y, 2, 0, 0, 0, 0, 200, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Jan-17, SIN-HND 2010-Jan-17, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Jan-17, SIN-HND 2010-Jan-17, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ12, 2010-Jan-18

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Jan-18, SIN-HND, 2010-Jan-18, 09:20:00, 2010-Jan-18, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Jan-18, SIN-HND 2010-Jan-18, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 2.14469e-319, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Jan-18, SIN-HND 2010-Jan-18, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Jan-18, SIN-HND 2010-Jan-18, Y, 2, 0, 0, 0, 0, 200, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Jan-18, SIN-HND 2010-Jan-18, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 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,
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,
SQ12 2010-Jan-19, SIN-HND 2010-Jan-19, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Jan-19, SIN-HND 2010-Jan-19, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ12, 2010-Jan-20

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity,
SQ12 2010-Jan-20, SIN-HND, 2010-Jan-20, 09:20:00, 2010-Jan-20, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

```
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Jan-20, SIN-HND 2010-Jan-20, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Jan-20, SIN-HND 2010-Jan-20, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Jan-20, SIN-HND 2010-Jan-20, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Jan-20, SIN-HND 2010-Jan-20, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Jan-20, SIN-HND 2010-Jan-20, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Jan-21
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Jan-21, SIN-HND, 2010-Jan-21, 09:20:00, 2010-Jan-21, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Jan-21, SIN-HND 2010-Jan-21, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Jan-21, SIN-HND 2010-Jan-21, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Jan-21, SIN-HND 2010-Jan-21, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Jan-21, SIN-HND 2010-Jan-21, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Jan-21, SIN-HND 2010-Jan-21, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Jan-22
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
```

```
SQL2 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,
SQL2 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,
SQL2 2010-Jan-22, SIN-HND 2010-Jan-22, Y, 1, 0, 0, 0, 0, 200, 0,
SQL2 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,
SQL2 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,
SQL2 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: SQL2, 2010-Jan-23
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 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,
SQL2 2010-Jan-23, SIN-HND 2010-Jan-23, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL2 2010-Jan-23, SIN-HND 2010-Jan-23, Y, 1, 0, 0, 0, 0, 200, 0,
SQL2 2010-Jan-23, SIN-HND 2010-Jan-23, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQL2 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,
SQL2 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: SQL2, 2010-Jan-24
*****
*****
Leg-Dates:
-----
```

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

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Jan-26, SIN-HND, 2010-Jan-26, 09:20:00, 2010-Jan-26, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Jan-26, SIN-HND 2010-Jan-26, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Jan-26, SIN-HND 2010-Jan-26, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Jan-26, SIN-HND 2010-Jan-26, Y, 2, 0, 0, 0, 0, 200, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Jan-26, SIN-HND 2010-Jan-26, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Jan-26, SIN-HND 2010-Jan-26, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ12, 2010-Jan-27

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Jan-27, SIN-HND, 2010-Jan-27, 09:20:00, 2010-Jan-27, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Jan-27, SIN-HND 2010-Jan-27, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Jan-27, SIN-HND 2010-Jan-27, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Jan-27, SIN-HND 2010-Jan-27, Y, 2, 0, 0, 0, 0, 200, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Jan-27, SIN-HND 2010-Jan-27, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Jan-27, SIN-HND 2010-Jan-27, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ12, 2010-Jan-28

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Jan-28, SIN-HND, 2010-Jan-28, 09:20:00, 2010-Jan-28, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Jan-28, SIN-HND 2010-Jan-28, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Jan-28, SIN-HND 2010-Jan-28, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Jan-28, SIN-HND 2010-Jan-28, Y, 2, 0, 0, 0, 0, 200, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Jan-28, SIN-HND 2010-Jan-28, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Jan-28, SIN-HND 2010-Jan-28, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ12, 2010-Jan-29

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Jan-29, SIN-HND, 2010-Jan-29, 09:20:00, 2010-Jan-29, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Jan-29, SIN-HND 2010-Jan-29, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Jan-29, SIN-HND 2010-Jan-29, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Jan-29, SIN-HND 2010-Jan-29, Y, 2, 0, 0, 0, 0, 200, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Jan-29, SIN-HND 2010-Jan-29, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Jan-29, SIN-HND 2010-Jan-29, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ12, 2010-Jan-30

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Jan-30, SIN-HND, 2010-Jan-30, 09:20:00, 2010-Jan-30, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Jan-30, SIN-HND 2010-Jan-30, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,

SQ12 2010-Jan-30, SIN-HND 2010-Jan-30, Y, 1, 0, 0, 0, 0, 200, 0,

SQ12 2010-Jan-30, SIN-HND 2010-Jan-30, Y, 2, 0, 0, 0, 0, 200, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,

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,

SQ12 2010-Jan-31, SIN-HND 2010-Jan-31, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,

SQ12 2010-Jan-31, SIN-HND 2010-Jan-31, Y, 1, 0, 0, 0, 0, 200, 0,

SQ12 2010-Jan-31, SIN-HND 2010-Jan-31, Y, 2, 0, 0, 0, 0, 200, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,

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,
SQ12 2010-Feb-01, SIN-HND 2010-Feb-01, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-01, SIN-HND 2010-Feb-01, Y, 1, 0, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-01, SIN-HND 2010-Feb-01, Y, 2, 0, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-01, SIN-HND 2010-Feb-01, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-01, SIN-HND 2010-Feb-01, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-02
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-02, SIN-HND, 2010-Feb-02, 09:20:00, 2010-Feb-02, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-02, SIN-HND 2010-Feb-02, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-02, SIN-HND 2010-Feb-02, Y, 1, 0, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-02, SIN-HND 2010-Feb-02, Y, 2, 0, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-02, SIN-HND 2010-Feb-02, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-02, SIN-HND 2010-Feb-02, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
```

FlightDate: SQ12, 2010-Feb-03

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-03, SIN-HND, 2010-Feb-03, 09:20:00, 2010-Feb-03, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-03, SIN-HND 2010-Feb-03, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-03, SIN-HND 2010-Feb-03, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-03, SIN-HND 2010-Feb-03, Y, 2, 0, 0, 0, 0, 200, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-03, SIN-HND 2010-Feb-03, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-03, SIN-HND 2010-Feb-03, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ12, 2010-Feb-04

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-04, SIN-HND, 2010-Feb-04, 09:20:00, 2010-Feb-04, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-04, SIN-HND 2010-Feb-04, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-04, SIN-HND 2010-Feb-04, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-04, SIN-HND 2010-Feb-04, Y, 2, 0, 0, 0, 0, 200, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-04, SIN-HND 2010-Feb-04, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-04, SIN-HND 2010-Feb-04, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ12, 2010-Feb-05

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-05, SIN-HND, 2010-Feb-05, 09:20:00, 2010-Feb-05, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-05, SIN-HND 2010-Feb-05, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-05, SIN-HND 2010-Feb-05, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-05, SIN-HND 2010-Feb-05, Y, 2, 0, 0, 0, 0, 200, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-05, SIN-HND 2010-Feb-05, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-05, SIN-HND 2010-Feb-05, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ12, 2010-Feb-06

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-06, SIN-HND, 2010-Feb-06, 09:20:00, 2010-Feb-06, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-06, SIN-HND 2010-Feb-06, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-06, SIN-HND 2010-Feb-06, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-06, SIN-HND 2010-Feb-06, Y, 2, 0, 0, 0, 0, 200, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-06, SIN-HND 2010-Feb-06, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-06, SIN-HND 2010-Feb-06, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

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

```
SQL2 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: SQL2, 2010-Feb-09
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 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,
SQL2 2010-Feb-09, SIN-HND 2010-Feb-09, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL2 2010-Feb-09, SIN-HND 2010-Feb-09, Y, 1, 0, 0, 0, 0, 200, 0,
SQL2 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,
SQL2 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,
SQL2 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: SQL2, 2010-Feb-10
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 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,
SQL2 2010-Feb-10, SIN-HND 2010-Feb-10, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL2 2010-Feb-10, SIN-HND 2010-Feb-10, Y, 1, 0, 0, 0, 0, 200, 0,
SQL2 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,
```

```
SQL2 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,
SQL2 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: SQL2, 2010-Feb-11
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 2010-Feb-11, SIN-HND, 2010-Feb-11, 09:20:00, 2010-Feb-11, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL2 2010-Feb-11, SIN-HND 2010-Feb-11, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL2 2010-Feb-11, SIN-HND 2010-Feb-11, Y, 1, 0, 0, 0, 0, 0, 200, 0,
SQL2 2010-Feb-11, SIN-HND 2010-Feb-11, Y, 2, 0, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQL2 2010-Feb-11, SIN-HND 2010-Feb-11, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQL2 2010-Feb-11, SIN-HND 2010-Feb-11, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL2, 2010-Feb-12
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 2010-Feb-12, SIN-HND, 2010-Feb-12, 09:20:00, 2010-Feb-12, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL2 2010-Feb-12, SIN-HND 2010-Feb-12, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL2 2010-Feb-12, SIN-HND 2010-Feb-12, Y, 1, 0, 0, 0, 0, 0, 200, 0,
SQL2 2010-Feb-12, SIN-HND 2010-Feb-12, Y, 2, 0, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
```

```
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-12, SIN-HND 2010-Feb-12, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-12, SIN-HND 2010-Feb-12, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-13
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-13, SIN-HND, 2010-Feb-13, 09:20:00, 2010-Feb-13, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-13, SIN-HND 2010-Feb-13, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-13, SIN-HND 2010-Feb-13, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-13, SIN-HND 2010-Feb-13, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-13, SIN-HND 2010-Feb-13, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-13, SIN-HND 2010-Feb-13, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-14
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-14, SIN-HND, 2010-Feb-14, 09:20:00, 2010-Feb-14, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-14, SIN-HND 2010-Feb-14, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-14, SIN-HND 2010-Feb-14, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-14, SIN-HND 2010-Feb-14, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
```

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-14, SIN-HND 2010-Feb-14, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-14, SIN-HND 2010-Feb-14, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ12, 2010-Feb-15

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-15, SIN-HND, 2010-Feb-15, 09:20:00, 2010-Feb-15, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-15, SIN-HND 2010-Feb-15, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,

SQ12 2010-Feb-15, SIN-HND 2010-Feb-15, Y, 1, 0, 0, 0, 0, 200, 0,

SQ12 2010-Feb-15, SIN-HND 2010-Feb-15, Y, 2, 0, 0, 0, 0, 200, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-15, SIN-HND 2010-Feb-15, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-15, SIN-HND 2010-Feb-15, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ12, 2010-Feb-16

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-16, SIN-HND, 2010-Feb-16, 09:20:00, 2010-Feb-16, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-16, SIN-HND 2010-Feb-16, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,

SQ12 2010-Feb-16, SIN-HND 2010-Feb-16, Y, 1, 0, 0, 0, 0, 200, 0,

SQ12 2010-Feb-16, SIN-HND 2010-Feb-16, Y, 2, 0, 0, 0, 0, 200, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-16, SIN-HND 2010-Feb-16, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-16, SIN-HND 2010-Feb-16, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ12, 2010-Feb-17

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-17, SIN-HND, 2010-Feb-17, 09:20:00, 2010-Feb-17, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-17, SIN-HND 2010-Feb-17, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,

SQ12 2010-Feb-17, SIN-HND 2010-Feb-17, Y, 1, 0, 0, 0, 0, 200, 0,

SQ12 2010-Feb-17, SIN-HND 2010-Feb-17, Y, 2, 0, 0, 0, 0, 200, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,

SQ12 2010-Feb-17, SIN-HND 2010-Feb-17, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

SQ12 2010-Feb-17, SIN-HND 2010-Feb-17, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ12, 2010-Feb-18

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity

SQ12 2010-Feb-18, SIN-HND, 2010-Feb-18, 09:20:00, 2010-Feb-18, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,

SQ12 2010-Feb-18, SIN-HND 2010-Feb-18, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,

SQ12 2010-Feb-18, SIN-HND 2010-Feb-18, Y, 1, 0, 0, 0, 0, 200, 0,

SQ12 2010-Feb-18, SIN-HND 2010-Feb-18, Y, 2, 0, 0, 0, 0, 200, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-18, SIN-HND 2010-Feb-18, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-18, SIN-HND 2010-Feb-18, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ12, 2010-Feb-19

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-19, SIN-HND, 2010-Feb-19, 09:20:00, 2010-Feb-19, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-19, SIN-HND 2010-Feb-19, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-19, SIN-HND 2010-Feb-19, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-19, SIN-HND 2010-Feb-19, Y, 2, 0, 0, 0, 0, 200, 0,

Subclasses:

Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-19, SIN-HND 2010-Feb-19, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-19, SIN-HND 2010-Feb-19, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,

FlightDate: SQ12, 2010-Feb-20

Leg-Dates:

Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-20, SIN-HND, 2010-Feb-20, 09:20:00, 2010-Feb-20, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,

LegCabins:

Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-20, SIN-HND 2010-Feb-20, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,

Buckets:

Flight, Leg, Cabin, Yield, AU/SI, SS, AV,

SegmentCabins:

Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-20, SIN-HND 2010-Feb-20, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-20, SIN-HND 2010-Feb-20, Y, 2, 0, 0, 0, 0, 200, 0,

```
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-20, SIN-HND 2010-Feb-20, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-20, SIN-HND 2010-Feb-20, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-21
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-21, SIN-HND, 2010-Feb-21, 09:20:00, 2010-Feb-21, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-21, SIN-HND 2010-Feb-21, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-21, SIN-HND 2010-Feb-21, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-21, SIN-HND 2010-Feb-21, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-21, SIN-HND 2010-Feb-21, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-21, SIN-HND 2010-Feb-21, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-22
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-22, SIN-HND, 2010-Feb-22, 09:20:00, 2010-Feb-22, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
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,
```

```
SQL2 2010-Feb-22, SIN-HND 2010-Feb-22, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQL2 2010-Feb-22, SIN-HND 2010-Feb-22, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQL2 2010-Feb-22, SIN-HND 2010-Feb-22, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL2, 2010-Feb-23
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 2010-Feb-23, SIN-HND, 2010-Feb-23, 09:20:00, 2010-Feb-23, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL2 2010-Feb-23, SIN-HND 2010-Feb-23, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL2 2010-Feb-23, SIN-HND 2010-Feb-23, Y, 1, 0, 0, 0, 0, 200, 0,
SQL2 2010-Feb-23, SIN-HND 2010-Feb-23, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQL2 2010-Feb-23, SIN-HND 2010-Feb-23, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQL2 2010-Feb-23, SIN-HND 2010-Feb-23, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQL2, 2010-Feb-24
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 2010-Feb-24, SIN-HND, 2010-Feb-24, 09:20:00, 2010-Feb-24, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQL2 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,
```

```
SQL2 2010-Feb-24, SIN-HND 2010-Feb-24, Y, 1, 0, 0, 0, 0, 200, 0,
SQL2 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,
SQL2 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,
SQL2 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: SQL2, 2010-Feb-25
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 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,
SQL2 2010-Feb-25, SIN-HND 2010-Feb-25, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQL2 2010-Feb-25, SIN-HND 2010-Feb-25, Y, 1, 0, 0, 0, 0, 200, 0,
SQL2 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,
SQL2 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,
SQL2 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: SQL2, 2010-Feb-26
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQL2 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,
SQL2 2010-Feb-26, SIN-HND 2010-Feb-26, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
```

```
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-26, SIN-HND 2010-Feb-26, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-26, SIN-HND 2010-Feb-26, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-26, SIN-HND 2010-Feb-26, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-26, SIN-HND 2010-Feb-26, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-27
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-27, SIN-HND, 2010-Feb-27, 09:20:00, 2010-Feb-27, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-27, SIN-HND 2010-Feb-27, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-27, SIN-HND 2010-Feb-27, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-27, SIN-HND 2010-Feb-27, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-27, SIN-HND 2010-Feb-27, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-27, SIN-HND 2010-Feb-27, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****
*****
FlightDate: SQ12, 2010-Feb-28
*****
*****
Leg-Dates:
-----
Flight, Leg, BoardDate, BoardTime, OffDate, OffTime, Date Offset, Time Offset, Elapsed, Distance, Capacity
SQ12 2010-Feb-28, SIN-HND, 2010-Feb-28, 09:20:00, 2010-Feb-28, 12:00:00, 07:40:00, 0, -05:00:00, 6300, 0,
*****
*****
LegCabins:
-----
Flight, Leg, Cabin, OffedCAP, PhyCAP, RgdADJ, AU, UPR, SS, Staff, WL, Group, CommSpace, AvPool, Avl, NAV,
SQ12 2010-Feb-28, SIN-HND 2010-Feb-28, Y, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 200, 9, 0, 0, 0, 0, 0,
*****
*****
Buckets:
-----
Flight, Leg, Cabin, Yield, AU/SI, SS, AV,
*****
*****
SegmentCabins:
```

```

-----
Flight, Segment, Cabin, FF, Bkgs, MIN, UPR, CommSpace, AvPool, BP,
SQ12 2010-Feb-28, SIN-HND 2010-Feb-28, Y, 1, 0, 0, 0, 0, 200, 0,
SQ12 2010-Feb-28, SIN-HND 2010-Feb-28, Y, 2, 0, 0, 0, 0, 200, 0,
*****
*****
Subclasses:
-----
Flight, Segment, Cabin, FF, Subclass, MIN/AU (Prot), Nego, NS%, OB%, Bkgs, GrpBks (pdg), StfBkgs, WLBkgs,
SQ12 2010-Feb-28, SIN-HND 2010-Feb-28, Y, 1, Y, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
SQ12 2010-Feb-28, SIN-HND 2010-Feb-28, Y, 2, M, 200 (0), 0, 0, 0, 0, 0 (0), 0, 0, 0, 0, 0, 0,
*****

```

12.11.6 Exploring the Predefined BOM Tree

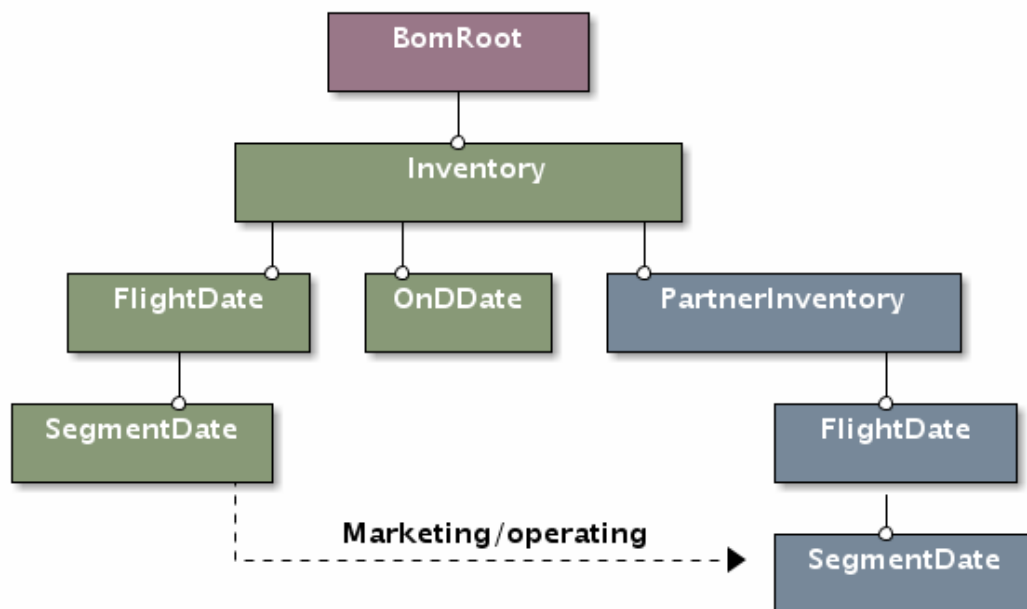


Figure 2: SimCRS BOM tree

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

12.11.6.1 Airline Network BOM Tree

- SIMCRS::ReachableUniverse
- SIMCRS::OriginDestinationSet
- SIMCRS::SegmentPathPeriod

12.11.6.2 Airline Schedule BOM Tree

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

12.11.7 Extending the BOM Tree

12.11.8 The travel solution calculation procedure

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

12.12 Supported Systems

12.12.1 Table of Contents

- [Introduction](#)
- [.1 SimCRS 0.1.x.1](#)
 - [Linux Systems](#)
 - * [Fedora Core 4 with ATLAS](#)
 - * [Gentoo Linux with ACML](#)
 - * [Gentoo Linux with ATLAS](#)
 - * [Gentoo Linux with MKL](#)
 - * [Gentoo Linux with NetLib's BLAS and LAPACK](#)
 - * [Red Hat Enterprise Linux with SimCRS External](#)
 - * [SUSE Linux 10.0 with NetLib's BLAS and LAPACK](#)
 - * [SUSE Linux 10.0 with MKL](#)
 - [Windows Systems](#)
 - * [Microsoft Windows XP with Cygwin](#)
 - * [Microsoft Windows XP with Cygwin and ATLAS](#)
 - * [Microsoft Windows XP with Cygwin and ACML](#)
 - * [Microsoft Windows XP with MinGW, MSYS and ACML](#)
 - * [Microsoft Windows XP with MinGW, MSYS and SimCRS External](#)
 - * [Microsoft Windows XP with MS Visual C++ and Intel MKL](#)
 - [Unix Systems](#)
 - * [SunOS 5.9 with SimCRS External](#)
- [SimCRS 3.9.1](#)
- [SimCRS 3.9.0](#)
- [SimCRS 3.8.1](#)

12.12.2 Introduction

This page is intended to provide a list of SimCRS supported systems, i.e. the systems on which configuration, installation and testing process of the SimCRS library has been successful. Results are grouped based on minor release number. Therefore, only the latest tests for bug-fix releases are included. Besides, the information on this page is divided into sections dependent on the operating system.

Where necessary, some extra information is given for each tested configuration, e.g. external libraries installed, configuration commands used, etc.

If you manage to compile, install and test the SimCRS library on a system not mentioned below, please let us know, so we could update this database.

12.13 SimCRS Supported Systems (Previous Releases)

- **Platform:** Intel Pentium 4
- **Operating System:** Fedora Core 4 (x86)
- **Compiler:** g++ (GCC) 4.0.2 20051125
- **SimCRS release:** 0.1.0
- **External Libraries:** From FC4 distribution:

```
- fftw3.i386-3.0.1-3
- fftw3-devel.i386-3.0.1-3
- atlas-sse2.i386-3.6.0-8.fc4
- atlas-sse2-devel.i386-3.6.0-8.fc4
- blas.i386-3.0-35.fc4
- lapack.i386-3.0-35.fc4
```

- **Tests Status:** All tests PASSED
- **Comments:** SimCRS configured with:

```
% CXXFLAGS="-O3 -pipe -march=pentium4" ./configure
```

- **Date:** March 7, 2006
- **Tester:** Tony Ottosson

- **Platform:** AMD Sempron 3000+
- **Operating System:** Gentoo Linux 2006.0 (x86 arch)
- **Compiler(s):** g++ (GCC) 3.4.5
- **SimCRS release:** 0.1.1
- **External Libraries:** Compiled and installed from portage tree:

```
- sci-libs/acml-3.0.0
```

- **Tests Status:** All tests PASSED
- **Comments:** BLAS and LAPACK libs set by using the following system commands:

```
% eselect blas set ACML
% eselect lapack set ACML
```

SimCRS configured with:

```
% export CPPFLAGS="-I/usr/include/acml"
% ./configure --with-blas="-lblas"
```

- **Date:** March 31, 2006
- **Tester:** Adam Piatyszek (ediap)
- **Platform:** Intel Pentium M Centrino
- **Operating System:** Gentoo Linux 2006.0 (x86)
- **Compiler:** g++ (GCC) 3.4.5
- **SimCRS release:** 0.1.1
- **External Libraries:** Compiled and installed from portage tree:
 - sci-libs/fftw-3.1
 - sci-libs/blas-atlas-3.6.0-r1
 - sci-libs/lapack-atlas-3.6.0
- **Tests Status:** All tests PASSED
- **Comments:** BLAS and LAPACK libs set by using the following system commands:

```
% eselect blas set ATLAS
% eselect lapack set ATLAS
```

SimCRS configured with:

```
% ./configure --with-blas="-lblas"
```

- **Date:** March 31, 2006
- **Tester:** Adam Piatyszek (ediap)
- **Platform:** Intel Pentium M Centrino
- **Operating System:** Gentoo Linux 2006.0 (x86 arch)
- **Compiler:** g++ (GCC) 3.4.5
- **SimCRS release:** 0.1.0
- **External Libraries:** Intel Math Kernel Library (MKL) 8.0.1 installed manually in the following directory: /opt/intel/mkl/8.0.1
- **Tests Status:** All tests PASSED
- **Comments:** SimCRS configured using the following commands:

```
% export LDFLAGS="-L/opt/intel/mkl/8.0.1/lib/32"
% export CPPFLAGS="-I/opt/intel/mkl/8.0.1/include"
% ./configure
```

- **Date:** February 28, 2006
- **Tester:** Adam Piatyszek (ediap)
- **Platform:** Intel Pentium M Centrino
- **Operating System:** Gentoo Linux 2006.0 (x86)
- **Compiler:** g++ (GCC) 3.4.5
- **SimCRS release:** 0.1.1
- **External Libraries:** Compiled and installed from portage tree:
 - sci-libs/fftw-3.1
 - sci-libs/blas-reference-19940131-r2
 - sci-libs/cblas-reference-20030223
 - sci-libs/lapack-reference-3.0-r2
- **Tests Status:** All tests PASSED
- **Comments:** BLAS and LAPACK libs set by using the following system commands:

```
% blas-config reference
% lapack-config reference
```

SimCRS configured with:

```
% ./configure --with-blas="-lblas"
```

- **Date:** March 31, 2006
- **Tester:** Adam Piatyszek (ediap)
- **Platform:** Intel Pentium 4
- **Operating System:** Red Hat Enterprise Linux AS release 4 (Nahant Update 2)
- **Compiler:** g++ (GCC) 3.4.4 20050721 (Red Hat 3.4.4-2)
- **SimCRS release:** 0.1.0
- **External Libraries:** BLAS, CBLAS, LAPACK and FFTW libraries from SimCRS External 2.1.1 package
- **Tests Status:** All tests PASSED
- **Date:** March 7, 2006
- **Tester:** Erik G. Larsson
- **Platform:** Intel Pentium 4 CPU 3.20GHz (64-bit)
- **Operating System:** SUSE Linux 10.0 (x86_64)
- **Compiler(s):** g++ (GCC) 4.0.2
- **SimCRS release:** 0.1.0

- **External Libraries:** BLAS, LAPACK and FFTW libraries installed from OpenSuse 10.0 RPM repository:

- blas-3.0-926
 - lapack-3.0-926
 - fftw3-3.0.1-114
 - fftw3-threads-3.0.1-114
 - fftw3-devel-3.0.1-114

- **Tests Status:** All tests PASSED

- **Comments:** SimCRS configured with:

```
% export CXXFLAGS="-m64 -march=nocona -O3 -pipe"  
% ./configure --with-lapack="/usr/lib64/liblapack.so.3"
```

- **Date:** March 1, 2006

- **Tester:** Adam Piatyszek (ediap)

- **Platform:** Intel Pentium 4 CPU 3.20GHz (64-bit)

- **Operating System:** SUSE Linux 10.0 (x86_64)

- **Compiler(s):** g++ (GCC) 4.0.2

- **SimCRS release:** 0.1.0

- **External Libraries:** Intel Math Kernel Library (MKL) 8.0.1 installed manually in the following directory: /opt/intel/mkl/8.0.1

- **Tests Status:** All tests PASSED

- **Comments:** SimCRS configured with:

```
% export CXXFLAGS="-m64 -march=nocona -O3 -pipe"  
% export LDFLAGS="-L/opt/intel/mkl/8.0.1/lib/em64t"  
% export CPPFLAGS="-I/opt/intel/mkl/8.0.1/include"  
% ./configure
```

- **Date:** March 1, 2006

- **Tester:** Adam Piatyszek (ediap)

- **Platform:** AMD Sempron 3000+

- **Operating System:** Microsoft Windows XP SP2, Cygwin 1.5.19-4

- **Compiler(s):** g++ (GCC) 3.4.4 (cygming special)

- **SimCRS release:** 0.1.1

- **External Libraries:** Installed from Cygwin's repository:

- fftw-3.0.1-2
 - fftw-dev-3.0.1-1
 - lapack-3.0-4

- **Tests Status:** All tests PASSED
- **Comments:** Only static library can be built. SimCRS configured with:

```
% ./configure
```

- **Date:** March 31, 2006
- **Tester:** Adam Piatyszek (ediap)
- **Platform:** AMD Sempron 3000+
- **Operating System:** Microsoft Windows XP SP2, Cygwin 1.5.19-4
- **Compiler(s):** g++ (GCC) 3.4.4 (cygming special)
- **SimCRS release:** 0.1.1
- **External Libraries:** Installed from Cygwin's repository:

```
- fftw-3.0.1-2  
- fftw-dev-3.0.1-1
```

ATLAS BLAS and LAPACK libraries from SimCRS External 2.1.1 package configured using:

```
% ./configure --enable-atlas --disable-fftw
```

- **Tests Status:** All tests PASSED
- **Comments:** Only st

12.13.1 SimCRS 3.9.1

12.13.2 SimCRS 3.9.0

12.13.3 SimCRS 3.8.1

12.14 Tutorials

12.14.1 Table of Contents

- [Preparing the AirSched Project for Development](#)
- [Your first networkBuilde](#)
 - [Summary of the different steps](#)
 - [Result of the Batch Program](#)
- [Network building with an input file](#)
 - [How to build a network input file?](#)
 - [Building the BOM tree with an input file](#)
 - [Result of the Batch Program](#)

12.14.2 Preparing the AirSched Project for Development

The source code for these examples can be found in the `batches` and `test/airsched` directories. They are compiled along with the rest of the `AirSched` project. See the [Users Guide](#) for more details on how to build the `AirSched` project.

12.14.3 Your first networkBuilder

12.14.3.1 Summary of the different steps All the steps below can be found in the same order in the batch `AirSched.cpp` program.

First, we instantiate the `AIRSCHEM_Service` object:

Then, we construct a default sample list of travel solutions and a default booking request (as mentioned in `ug_procedure_bookingrequest` and `ug_procedure_travelsolution` parts):

For basic use, the default BOM tree can be built using:

The main step is the network building (see [The travel solution calculation procedure](#)):

12.14.3.2 Result of the Batch Program When the `AirSched.cpp` program is run (with the `-b` option), the log output file should look like:

What is interesting is to compare the travel solution list (here reduced to a single travel solution) displayed before:

and after the network building:

Between the two groups of dashes, we can see that a network option structure has been added by the network builder: the price is 450 EUR for the Y class, the ticket is refundable but there are exchange fees and the customer must stay over on saturday night.

Let's return to our default BOM tree display: the only network rule stored was a match for the travel solution into consideration (same origin airport, same destination airport, flight date included in the network rule date range, same airline "BA", ...).

By looking at the network rule trip type "RT", we can guess we face a round trip network: that means the price given in the default bom tree construction in `stdair::CmdBomManager.hpp` has been divided by 2 because we are considering either an inbound trip or an outbound one.

12.14.4 Network building with an input file

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

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

We deduce:

- we need a network rule whose origin-destination couple is "LHR, SYD".
- the date range must include the date "2011-06-10".
- the time range must include the time "21:45".
- the airline operating is "BA", so it must be the airline pricing.

We can deduce a part of our network rule file :

We have no information about stay duration and advance purchase (such information are contained into the booking request): so let us put "0" to embrace all the requests possible.

No information for the point-of-sale and the channel too: let us consider all the channels ("IN", "DN", "IF" and "DF") and all the points of sale (the origin "LHR", the destination "SYD" and the rest-of-the-world "ROW") existing. To access this information, we could look into the default booking request.

The input file is now:

Let us say we have just the Economy cabin "Y" and British Airways prices ticket for class "Y".

No information about the trip type, so we duplicate all the network rules for both type: one-way "OW" and round-trip "RT" (to access this information, we could look to the default booking request).

The network options are all set to a default value "T" (meaning true) and the network values are chosen to be all distinct.

We obtain:

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

12.14.4.3 Result of the Batch Program When the `AirSched.cpp` program is run with the `-f` option linking with the file built just above:

```
~/AirSched -f ~/<YourFileName>.csv
```

the last lines of the log output should look like:

```
[D]~/AirSchedgit/AirSched/batches/AirSched.cpp:223: Travel solutions:
  [0] [0] BA, 9, 2011-06-10, LHR, SYD, 21:45 --- Y, 145, 1 1 1 ---
```

We have just one network option added to the travel solution. We can deduce from the price value 145 that the network builder used the network rule number 15 to price the travel solution. We have an inbound or outbound trip of a round trip: the total price 290 has been divided by 2.

12.15 Command-Line Test to Demonstrate How To Test the SimCRS Project

```

*/
// //////////////////////////////////////
// Import section
// //////////////////////////////////////
// STL
#include <sstream>
#include <fstream>
#include <string>
#include <cmath>
// Boost Unit Test Framework (UTF)
#define BOOST_TEST_DYN_LINK
#define BOOST_TEST_MAIN
#define BOOST_TEST_MODULE CRSTestSuite
#include <boost/test/unit_test.hpp>
// StdAir
#include <stdair/basic/BasLogParams.hpp>
#include <stdair/basic/BasDBParams.hpp>
#include <stdair/basic/BasFileMgr.hpp>
#include <stdair/bom/TravelSolutionStruct.hpp>
#include <stdair/bom/BookingRequestStruct.hpp>
#include <stdair/service/Logger.hpp>
// SimFQT
#include <simfqt/SIMFQT_Types.hpp>
// SimCRS
#include <simcrs/SIMCRS_Service.hpp>
#include <simcrs/config/simcrs-paths.hpp>

namespace boost_utf = boost::unit_test;

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

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

    ~UnitTestConfig() {
    }
};

// //////////////////////////////////////
const unsigned int testSimCRSHelper (const unsigned short iTestFlag,
```

```

const stdair::Filename_T& iScheduleInputFilename,
const stdair::Filename_T& iOnDInputFilename,
const stdair::Filename_T& iFRAT5InputFilename,
const stdair::Filename_T& iFFDisutilityInputFilename,
const stdair::Filename_T& iYieldInputFilename,
const stdair::Filename_T& iFareInputFilename,
const bool isBuiltin,
const unsigned int iExpectedNbOfTravelSolutions,
const unsigned int iExpectedPrice) {

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

// Output log File
std::ostream oStr;
oStr << "CRSTestSuite_" << iTestFlag << ".log";
const stdair::Filename_T lLogFilename (oStr.str());

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

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

stdair::Date_T lPreferredDepartureDate;;
stdair::Date_T lRequestDate;
stdair::TripType_T lTripType;

// Check whether or not a (CSV) input file should be read
if (isBuiltin == true) {

    // Build the default sample BOM tree
    simcrsService.buildSampleBom();

    lPreferredDepartureDate = boost::gregorian::from_string ("2010/02/08");
    lRequestDate = boost::gregorian::from_string ("2010/01/21");
    lTripType = "OW";

} else {

    // Build the BOM tree from parsing input files
    stdair::ScheduleFilePath lScheduleFilePath (iScheduleInputFilename);
    stdair::ODFilePath lODFilePath (iOnDInputFilename);
    stdair::FRAT5FilePath lFRAT5FilePath (iFRAT5InputFilename);
    stdair::FFDisutilityFilePath lFFDisutilityFilePath (iFFDisutilityInputFilename);
    const SIMFQT::FareFilePath lFareFilePath (iFareInputFilename);
    const AIRRAC::YieldFilePath lYieldFilePath (iYieldInputFilename);
    simcrsService.parseAndLoad (lScheduleFilePath, lODFilePath,
                                lFRAT5FilePath, lFFDisutilityFilePath,
                                lYieldFilePath, lFareFilePath);

    lPreferredDepartureDate = boost::gregorian::from_string ("2011/01/31");
    lRequestDate = boost::gregorian::from_string ("2011/01/22");
    lTripType = "RI";
}

// Create an empty booking request structure
const stdair::AirportCode_T lOrigin ("SIN");
const stdair::AirportCode_T lDestination ("BKK");
const stdair::AirportCode_T lPOS ("SIN");
const stdair::Duration_T lRequestTime (boost::posix_time::hours(10));
const stdair::DateTime_T lRequestDateTime (lRequestDate, lRequestTime);
const stdair::CabinCode_T lPreferredCabin ("Eco");

```

```

const stdair::PartySize_T lPartySize (3);
const stdair::ChannelLabel_T lChannel ("IN");
const stdair::DayDuration_T lStayDuration (7);
const stdair::FrequentFlyer_T lFrequentFlyerType ("M");
const stdair::Duration_T lPreferredDepartureTime (boost::posix_time::hours(10));
const stdair::WTP_T lWTP (1000.0);
const stdair::PriceValue_T lValueOfTime (100.0);
const stdair::ChangeFees_T lChangeFees (true);
const stdair::Disutility_T lChangeFeeDisutility (50);
const stdair::NonRefundable_T lNonRefundable (true);
const stdair::Disutility_T lNonRefundableDisutility (50);
const stdair::BookingRequestStruct lBookingRequest (lOrigin, lDestination,
                                                    lPOS,
                                                    lPreferredDepartureDate,
                                                    lRequestDateTime,
                                                    lPreferredCabin,
                                                    lPartySize, lChannel,
                                                    lTripType, lStayDuration,
                                                    lFrequentFlyerType,
                                                    lPreferredDepartureTime,
                                                    lWTP, lValueOfTime,
                                                    lChangeFees,
                                                    lChangeFeeDisutility,
                                                    lNonRefundable,
                                                    lNonRefundableDisutility);

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

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

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

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

BOOST_CHECK_EQUAL (lNbOfTravelSolutions, iExpectedNbOfTravelSolutions);

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

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

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

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

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

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

BOOST_CHECK_MESSAGE (std::floor (lFareOption.getFare() + 0.5)

```

```

        == iExpectedPrice, oMessageKeptFare.str());

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

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

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

return isSellSuccessful;
}

// ////////////////////////////////// Main: Unit Test Suite //////////////////////////////////

// Set the UTF configuration (re-direct the output to a specific file)
BOOST_GLOBAL_FIXTURE (UnitTestFixture);

// Start the test suite
BOOST_AUTO_TEST_SUITE (master_test_suite)

BOOST_AUTO_TEST_CASE (simcrs_simple_simulation_test) {

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

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

    // FRAT5 curve input file name
    const stdair::Filename_T lFRAT5InputFilename (STDAIR_SAMPLE_DIR
        "/frat5.csv");

    // Fare family disutility curve input file name
    const stdair::Filename_T lFFDisutilityInputFilename (STDAIR_SAMPLE_DIR
        "/ffDisutility.csv");

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

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

    // State whether the BOM tree should be built-in or parsed from input files
    const bool isBuiltin = false;

    const unsigned int lExpectedPrice = 400;
    const unsigned int lExpectedNbOfTravelSolutions = 1;

    bool isSellSuccessful = false;

    BOOST_CHECK_NO_THROW (isSellSuccessful =
        testSimCRSHelper (0,
            lScheduleInputFilename,
            lOnDInputFilename,
            lFRAT5InputFilename,
            lFFDisutilityInputFilename,
            lYieldInputFilename,

```

```

        lFareInputFilename,
        isBuiltin,
        lExpectedNbOfTravelSolutions,
        lExpectedPrice));

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

BOOST_CHECK_EQUAL (isSellSuccessful, true);

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

}

BOOST_AUTO_TEST_CASE (simcrs_simple_default_bom_simulation_test) {

    // State whether the BOM tree should be built-in or parsed from input files
    const bool isBuiltin = true;

    const unsigned int lExpectedPrice = 900;
    const unsigned int lExpectedNbOfTravelSolutions = 1;

    bool isSellSuccessful = false;

    BOOST_CHECK_NO_THROW (isSellSuccessful =
        testSimCRSHelper (1,
            " ", " ", " ", " ", " ", " ", " ", " ",
            isBuiltin,
            lExpectedNbOfTravelSolutions,
            lExpectedPrice));

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

    BOOST_CHECK_EQUAL (isSellSuccessful, true);

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

}

// End the test suite
BOOST_AUTO_TEST_SUITE_END()

/*!
```