Oracle Real-Time Scheduler 1.13.2

Planner 10.5

Host Interface Manual
Revision 1.0

August 2010

Table of contents

Chapter 1: Introduction

The “Planner Host Interface Manual” is intended for users who wish to write an interactive software interface (called a Gateway) for communication between the Host System and Planner rather than operating in a batch (non-interactive) mode using HIP Files.

This manual includes:

- Detailed description of each of the Data objects and their properties.
- System limitations and data range values.
- Considerations of how to construct the data to match the specific scheduling task.
- Description of functional features of the system and the way to utilise them.
- Communication protocols and the mechanics of packet communication.
- How to control the flow of data to and from Planner.
- Programming examples in C to assist in the creation of Gateway software.

1.1. How To Use This Manual

Chapter 1: Introduction -explains the layout and content of the host interface Manual, introduces the reader to Planner and its view of the outside world, defines some commonly used terminology specific to Planner and illustrates the various Planner system schematics.

Chapter 2: Basic Planner Concepts -describes the data types that can be provided by the Host System for scheduling, and how Planner organises the basic data into meaningful arrangements (or schedule solutions). A brief description of Multiple Time Zone handing in Planner is also available in this chapter.

Chapter 3: Matching the Operation - describes some operational imperatives, how to set up the data structures to represent particular operational requirements, some operational scenarios and using/editing zonefiles.

Chapter 4: Interfacing to Planner - describes the Data Packets and the protocol required when communicating with Planner via user-written Gateway software.

Chapter 5: Host Database Considerations - describes host database changes that may have to be made to the Host Database tables to ensure gateway connectivity to Planner.

Chapter 6: Scheduling - describes how to use an interactive Gateway interface to accomplish the scheduling task both automatically and manually.

Chapter 5: Real Time Interfacing - describes the real-time environment for real-time interfacing.

Chapter 6: Mobile Data Messaging - describes communication to and from MDT(s) and Planner.

Chapter 7: Packets (10.5) - describes the functionalities supported by Planner packets, the system limits set for Planner Objects, the events handled by Planner, the Planner data definitions and packet descriptions.

Appendix A: - Programming Examples - illustrates the general concept of communicating with Planner.

1.2. WHAT IS PLANNER

Planner is the name of a Oracle Corporation software product that can be used to assist a Allocator to efficiently allocate loads to a fleet of vehicles for distribution to or collection from a number of different locations.

The Oracle Corporation product - **Planner** described in this manual and its companion Planner User Manual, is an efficient **Scheduler** designed for the transportation industry. In particular, Planner has been designed to -

- Optimise the solutions to warehouse distribution problems.
- Handle the problems that can arise for “Less than Truck Load” (LTL) deliveries.
- Schedule courier-type operations.

The Planner Scheduler is versatile and flexible. It can handle the scheduling of job deliveries that may be composed in many different ways and which may require delivery at arbitrary times. It can also readily handle (in **real time**) delivery variations caused by weather conditions or customer uncertainty. Planner makes the best possible use of the vehicles available to deliver the loads and specifies the most appropriate vehicles for the task at hand.

Planner may be set up to incorporate the “*business rules*” of the enterprise. These rules will be considered by Scheduler to create an operating schedule which is both efficient and practical.

The goal of Planner is to produce an operating schedule of *“least total cost*”.

Planner is at all times under the control of the Allocator(s) who can monitor the state of the schedule on a large screen display and control the operation using easy “point and click” mouse movements. The Planner Client interface display makes good use of colour and shape to convey all the required information to the Allocator(s) who can see at a glance the status of all the jobs, individual loads and vehicles. The display indicates the progress of each load from the time it is ordered until the time of delivery. It shows all relevant time marks - either predicted or actual. The layout of the Planner display can be configured to suit the individual requirements of each Allocator.

Geographic information is displayed on a map display window that shows the travel path and route information, some details of geography, and the current or predicted location of the vehicles. If it is available, GPS information from the vehicles can be used to periodically update the display of vehicle position.

Note: Please refer “Glossary” on page 1 - 6 for specific Planner terms used in the Manual.

1.3 PLANNER AND THE HOST SYSTEM

Interactive communication between the Host system and Planner is accomplished via a user-written Gateway interface.

The following chapters will describe how to interface to Planner.

1.3.1 What Is Scheduling?

Scheduling is the act of allocating STOPS (collection and/or distribution) to VEHICLES in an efficient manner to provide a solution to the transportation problem, taking into account all the constraints and limitations specified. Planner can accomplish this in three ways -

Automatic Scheduling

Manual Scheduling

Semi-Automatic Scheduling

In “Automatic Schedule” mode, Planner is a program which provides the “least cost” optimised solution to a transportation or scheduling problem according to a set of business and operating rules.

In “Manual Schedule” mode the solution is hand built, and this mode facilitates a rapid response to such questions as “What happens if I assign this STOP to that VEHICLE.”? Manual Scheduling can take place at any time, even during an Automatic Schedule - in which case the Manual changes may be overridden by Scheduler if it finds a better solution. To preserve the changes made during Manual Scheduling, Scheduler must be turned off.

“Semi-Automatic Scheduling” is a combination of the other two modes such that the Planner can build part of the schedule manually and “freeze” that solution, then use the “Automatic” mode to schedule any remaining allocations.

In any mode, the operation of Planner may be controlled and monitored using the computer’s display screen and mouse.

Once the initial Data Set for the problem under consideration has been loaded into Planner, any number of different solutions can be created to investigate the effects of varying the Cost Control parameters which Planner uses to control the scheduling.

Work schedule solutions can be created manually using Planner (if required), but Planner is a much more powerful tool when used in automatic or semi-automatic mode because it uses an optimising process to make the best use of time and resources while still implementing any business rules specified by the Allocator.

1.3.2 How Planner Sees the World

When ***Planner*** is purchased, a geographical region of operation will be agreed. When ***Planner*** is installed, it will be accompanied by an Oracle Corporation “map” containing the necessary data about the travel paths available in the designated geographical region.

A “MAP” consists of a list of NODES (geographical locations, suburbs etc.) and a list of inter-node LINKS (travel paths) whose direction, distance, and travel time are known.

The maps are generally limited to not more than 15000 non-trivial NODES.

The objectives of the map are -

- To provide **connectivity** (route) information
- To provide **travel time** information
- To provide **graphical display** information

If necessary, Oracle Corporation can provide a map maintenance software product called ***Geography Manager***^{TM}.

1.3.2.1 Map Co-ordinates

Planner uses maps that use the Latitude and Longitude coordinate system OR the Grid coordinate system. Co-ordinates are specified in decimal degrees using the normal cartographic conventions ie latitude -90.0 to +90.0 degrees, and longitude -180.0 to + 180.0 degrees. Grid coordinates are also specified in decimal degrees.

1.3.2.2 What’s in a Map?

The NODES and LINKS in a Oracle Corporation map are constructed to represent the locations and connectivity necessary to solve a particular transportation problem.

The NODES and LINKS are generally NOT A STREET MAP, although they work in a similar fashion in providing ***Planner*** with a detailed travel path (or route) from one point to another. This means that different transportation problems will need maps which could contain very different information.

Maps are also used to display other geographical information such as coastlines, rivers, and other features which help to identify the geographical region being displayed. These maps can be displayed simultaneously with the connectivity maps, but of course they are not used for travel calculations.

1.4 GLOSSARY

1.4.1 System

ERS -The **Oracle Real-Time Scheduler**package, including)amongst others the Planner and Resource Manager components

Planner - The automated scheduling component of ERS.

Resource Manager-DataBase component of the ERS, managing the definition of resources and general data-persistency.

1.4.1.1 Processes

Scheduler, SmAuto- The names given to the Planner process which maintains and optimizes a Schedule. Scheduler creates a schedule by recommending Job-to vehicle allocations while specifying the order in which vehicles should complete Jobs. A Scheduler may receive updates on existing Vehicles and Stores from an Allocator
Any changes received, from whichever source, are broadcast to all attached Clients.

Switch, Primary Switch, plannerSwitch- The names given to the Planner process which interfaces between Planner on one side, and the host system applications on the other. The Switch receives Vehicles, depots and Orders from the host system. A switch has the following tasks:
- Determine which process is Scheduler, which are Slot-generators.
- If necessary, promote one of the Slot-generators to Scheduler.
- Alternate Slot-requests between the available Slot-generators.
- Ensure no data is lost during failover.
- Ensure all appropriate connecting processes are kept up to date.
-Reload new Scheduler or Slot-generator.

Secondary Switch, plannerSwitch- Same as the Primary Switch, except that it is started by Primary Switch, and is kept up-to-date with all data and status information from Primary Switch at all time. It promotes itself to Primary Switch upon Switch Failover)

Slot-generator, SmAuto- Is same as the Planner Scheduler process, but with the exception that it does not create schedules. It uses updates from the Switch to maintain an up-to-date schedule. Receives Slot requests from the App Server process, to which it responds with a number of delivery Slots via the Switch.

GUI,mncGUI,Client,plannerCliect,Display- The name given to the Planner process which controls the display screen at which the Allocator works.

Launch Pad - Launch Pad is a graphical interface which helps in configuring and maintaining Planner. It facilitates selection of configuration files to use, version of planner to run and modification of the configuration files. It also runs and monitors all the planner processes.

Print Manifest Server-Print Manifest Server is a process with the ability to print centrally to feed a distribution system with Postscript files containing printed maps, manifests and turn by turn instructions.

GUI Server The GUI Server is a process which connects as a Client either to SmAuto or to another GUI Server while allowing multiple PlannerClient connections to itself.

1.4.1.2 Related Terms

Out-of-Sequence event-A Stop-status transition from one of the states on the left of table below, to one on the right, for a Stop which is (previously to the status change) not the first non ARRIVED+ Stop within the Shift.

FREE	ENROUTE
ASSIGNED	ARRIVED
ALLOCATED	STARTED
PENDING_DESPACHED	COMPLETED
DESPATCHED	
ACKNOWLEDGED	
UNDESPATCHED	
ENROUTE	
POSTPONED	

Host System - A computer system that controls the administration of the business enterprise.

Operator, Allocator, human scheduler- A person who will control and adjust the operation of Planner through the “Planner GUI” while optionally overriding the decisions made by “Planner Scheduler”.

Viewer **-**A person who may examine, but not modify, the schedule.

Schedule, solution Plan- The solution created by Planner for the problem specified.

MDTs - Mobile Data Terminals used to transmit messages to and receive messages from the driver.

1.4.2 Data Storage and Transfer

Database Table - A Table in a Relational Database Management System (RDBMS) maintained by the Host System. The Table will be a primary source of data to be scheduled and which can receive the scheduling solution(s) created by Planner.

Hip File - A file in ASCII format, local to Planner that can be used to store intermediate or temporary scheduling solutions. These files can be reloaded. (See HIP below)

HIP - Host Interface Protocol. An ASCII communications protocol between processes.

IDB - Internal Database. A memory resident data repository within Planner containing the state of the schedule as presented to the user.

Plan - The data set to be scheduled.

1.4.3 Planner Objects

Depot - A place where goods are stored in bulk.

Depot-stop - An object representing the collection or delivery of goods for one or more jobs during a single visit to a depot.

Depot Access Time Window- Time interval during which the depot can be accessed.

Depot Service Time Window- Agreed time interval during which access to the depot is preferred.

Product_Depot - details the available product and its capacity (in units) for a depot

Resource - Resource is a combination of vehicle and drivers (crew) that are expected to be available for working during a specified work period called Shift.

Vehicle - A load-carrying transporter known to Planner.

Shift - A combination of a Resource and a time period of availability with or without a set of defining attributes. Goods may remain on Resources between Shifts

Or

Contiguous period of time when a Crew is available to perform work.

Attribute - Resource Attribute.

Crew - Group of workers or Drivers working on the same Shift.

Driver - Entity who is able to carry out work. Usually a field service Engineer.

Proficiency - A defined hierarchy of levels of ability within a Skill. A higher proficiency level allows more difficult Jobs to be completed.

Resource Attribute- Describes an aspect of a Crew not related to Skill. Attributes are generally not derived from Crew-members, but indicate special equipment carried, or attributes of the Crew Vehicle. For example: HAS_PUMP, TAIL_LIFT.

Skill - The ability of a Crew or a Driver to perform a certain type of work.

Stop - Represents the execution of a short, well-defined Task, e.g. delivering goods at a certain address, collecting goods or providing some service.

Task - An amount of work to be performed at a single location. A general definition encompassing both Multi-Shift Tasks and Stops.

Haul - A number of consecutive shifts to be executed by the same resource. Consecutive Shifts are connected by a parameter which is carried forward from one Shift into the next shift.

Reserve Capacity -Time available/reserved in Shift to do additional Jobs.

EOTT - Engineer’s own travel time. That is the time the engineer travels from home to the first Job or from the last Job to his home on his own expense.

Stop - A place to be visited by a Resource. The visit may involve a Collection or Distribution. Special types of Stop are Depot, Logon, Logoff, and Break.

Stop Access Time Window- Time interval during which the stop can be serviced.

Stop Service Time Window- Agreed upon Time interval when servicing of the stop is preferred.

SLATW **-** Service Level Agreement time windows (for e.g. a courier may agree to arrive within 2 hours of booking).

AccessTW **-** There is no access to the site outside of the window (for e.g. shop opens from 09:00 to 17:00).

Break - Breaks represent rest periods for the driver. They are inserted at a specified time for a specified interval within a tour based on the break information provided initially to the Planner Scheduler

Job - One or more Stops grouped for transportation purposes. All the Stops in the Job must be completed by the same Resource.

Object - A collection of data representing a “real” concept, e.g., Stop, Depot, Resource.

Group - A number of Jobs grouped into a larger unit. Once a number of Jobs have been grouped, Scheduler can manipulate all Jobs in the Group in a single operation. Any single Job that has not been explicitly grouped is considered to be the only member of its own Group, that is, all “group” operations will apply to this Job

Route - Another name for “Haul”.

Trip, Tour - Consists of all the work being done by a Resource between log-on and log-off.

Real Object - An object for which there exists a packet representing all attributes, for example, a Stop, a Resource.

Virtual Object - An object constructed from more than one “Real Object”, for example, a Run, Haul or Group.

Run -Loading at depot, departure and subsequent delivery of all Jobs loaded. In other words a Run consists of a Depot Stop and all Stops for which goods are either collected from or dropped at the Depot Stop.

Round -Round is the diagonal measure estimated by Planner for a Run.

Round End Area -Round End Area is the rectangular geographic area estimated by Planner with the last stop of a Run at its geographic centre.

Closed Run -A run on which only manual operations are possible, i.e. Scheduler cannot change the set of Jobs associated with the run, nor can a Slot-generator suggest Slots that require a Job to be inserted into this run.

run-duration -On-board time of the last delivery of a run.

on-board time -The time between departure from the depot and the latest of arrival at the delivery address or start of the delivery time-window.

Slot -A pre-determined time window during which a delivery/pick-up can take place. Generally a day is subdivided into a number of delivery slots.

Remaining Capacity-The number of deliveries that may still be booked for a depot/day before the maximum capacity is reached.

Region -The geographical area covered by a business can be subdivided into number of independent regions for scheduling purposes. Each of these regions is defined by the Depots assigned to it.

Foreign Region **-** A Region which is not the Region handled by the process or Allocator performing the action.

depot capacity - The maximum capacity in terms of weight or volume that can be collected at or distributed from a Depot within a specific Depot Time Window.

Foreign depot - A depot which is not the depot handled by the process or Allocator performing the action.

1.4.4 Operations

Allocate - A Job is Allocated when it has been scheduled for a Haul or a Shift, and Planner has been restricted from removing this Job from the Haul or Shift.

Assign - A Job is Assigned when it has been scheduled for a Haul or a Shift, but Planner is not restricted from removing this Job from the Haul or Shift.

Close - A Run consisting of a number of Allocated Stops and a Depot Stop Run is Closed which implies Planner cannot insert, delete or reassign any of the Stops in the Run.

1.4.5 Host system Processes

Data Interface **-** Consists of one or more process which can do the following:
-handles interaction with the customer, and communicates with Planner through a GateWay. Provides Planner with Jobs and Slot-requests. Receives requested Slots from Planner.
-maintains details about Vehicles and Depots. Provides Planner (through GateWay) with information about Vehicles and Depot, including real-time updates received from the field.

Gateway - An interface process between the host system and Planner.

Database backup processes- reloads Planner with the data needed within the operational window, either at initial start-up, or after a total failure.
- loads Planner with data to define the next day available for Job slots/scheduling, i.e. today + 21 days.
- acts as an interface to the non volatile storage system. Maintains all Jobs and the plan data associated with them. Receives updates to the current state of the schedule from Planner, storing them in a database, where it can be retrieved by other Database backup processes.

1.4.6 Time Zone Terms

Absolute Time - An absolute time is a particular point in time i.e. 8am Monday 20th of October 2008 in Melbourne is an absolute time. It is completely unambiguous as to what this time refers to.

Abstract Time - Every Monday at 8am is an abstract time. An abstract will depend on the reference time zone. It will also depend on the time of year, i.e. daylight savings time.

UTC - Coordinated Universal Time, essentially equivalent to GMT Greenwich Mean Time.

Time Zone - A time zone is a region of the earth that has uniform standard time, usually referred to as the local time. By convention time zones compute their local time as an offset from UTC.

Time zones are divided in standard and daylight saving (or summer).

[The] names are all of the form Area/Location, where Area is the name of a continent or ocean, and Location is the name of a specific location within that region, usually cities or small islands. That is; Australia/Melbourne

Local Time - Also known as wall time, this is the time used in a particular location.

To disambiguate the display of local time the UTC offset is included in the display

That is; 02:39 UTC+10 for 2:30am local time which is offset from UTC by 10 hours. Or 15:23 UTC+5:30 for 3:23pm local time which is offset from UTC by 5 hours 30 minutes.

UTC can be replaced by a Z representing zero offset time zone, sometimes called zulu time.

1.4.7 Miscellaneous Terms

List - String containing a comma-separated set of object identifiers, e.g. “RED,WHITE,BLUE”.

Partial Specification-Partly specifies a selection, with the expectation that the recipient will return a list of matching complete specifications.

Synchronized -Two processes are “synchronized” when they represent the same version of the contents of some “data repository”.

FailOver **-** The process of recovery from a partial failure. The system can be restored to its original functionality, without loss of data, with minimal downtime.

Crash Recovery **-** The process of recovering from a total failure. The system must be fully reloaded from stored back-up data. Loss of data is likely.

Catastrophic Failure- Any failure FailOver cannot reliably cope with. A Catastrophic Failure requires a complete re-initialisation of the system, followed by a complete reload of data by DBLoad.

AB Request -Appointment Booking Request same as Slot Request

Initial AB Request-AB Request sent for a new Task.

Repeat AB Request-AB Request sent for an existing Task (Request for Re-Booking).

Conditional Booking -The operation to assign the Job to a compatible Shift in the schedule if the cost of doing so is below MAX_COST.The Task is retained by the system only if the process is successful, if not the Task is deleted.

Conditional Assignment -Same as Conditional Booking

Initial Conditional Booking-Conditional Booking sent for a new Task.

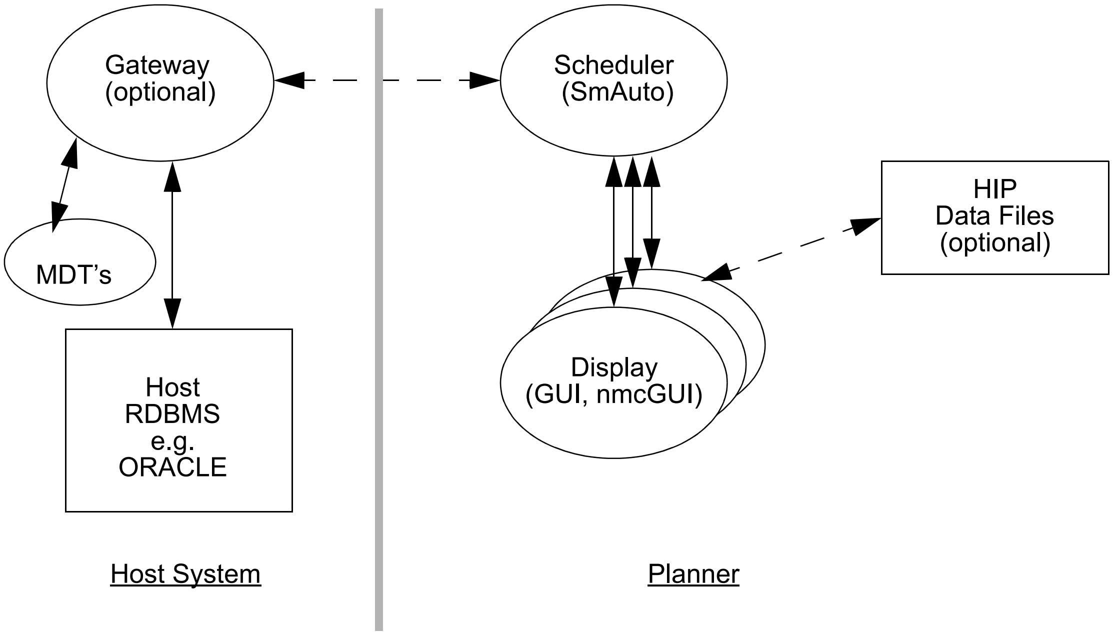
Repeat Conditional Booking -Conditional Booking send for an existing Task (Re-Booking).

~~Not Implemented~~ -Not Implemented.

1.5 Typical Planner System Schematic

1.5.1 Basic Planner architecture

Planner is made up of several independent software modules.

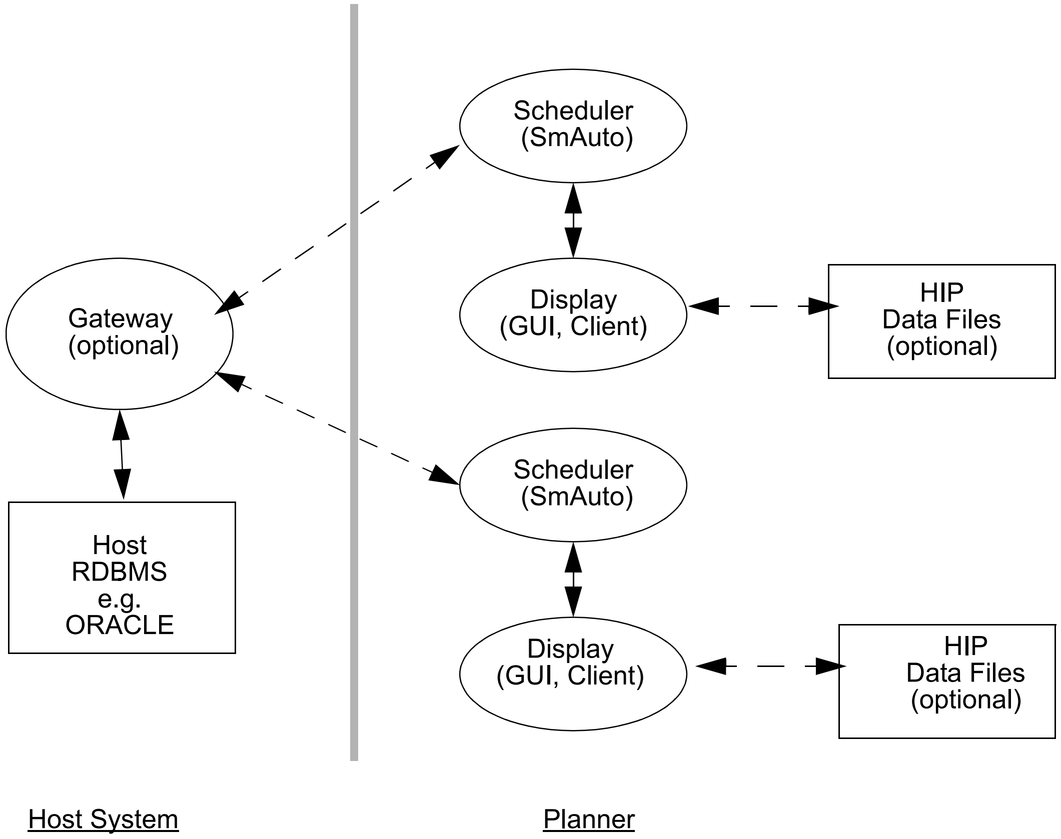


- Scheduler - process name SmAuto - compares the relative (notional) cost of all possible solutions.
- Display - process name GUI or nmcGUI - a Graphical User Interface that handles all screen display, keyboard and mouse activity.
- MDT's - Mobile Data Terminals to send messages to and receive messages from drivers/vehicles.
- Gateway - an optional module that interfaces with a Host System Database.

These processes will interact (optionally) with a Host System interface.

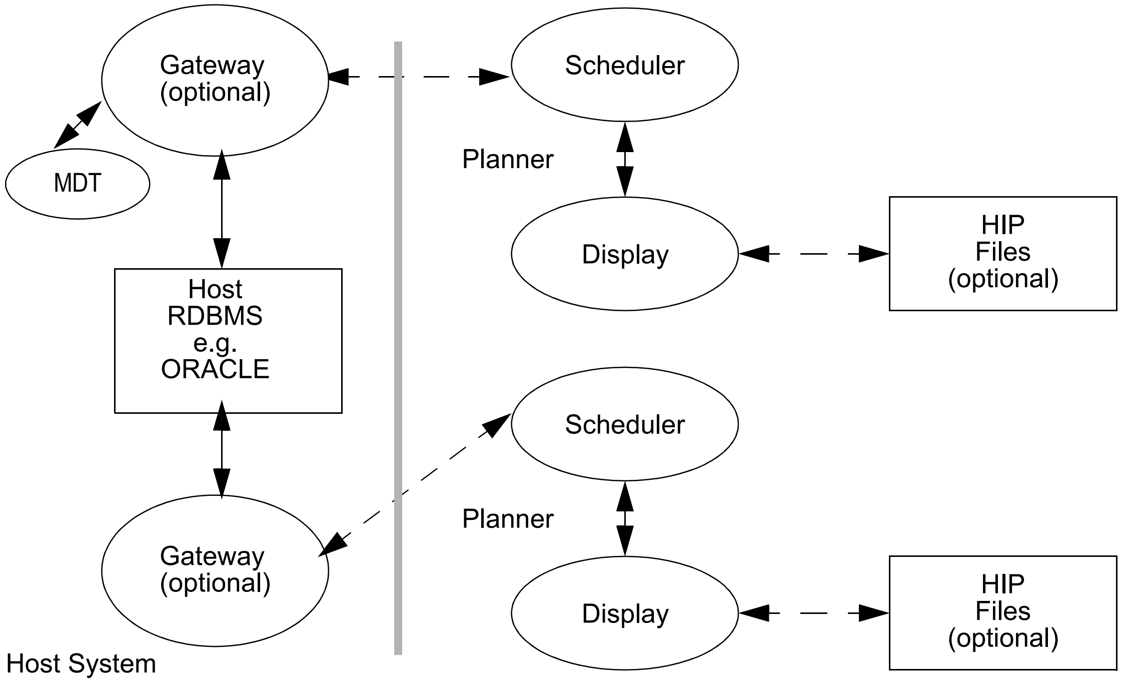
1.5.2 Planner Architecture for multiple regions

If more than one geographical region is to be scheduled from the same installation and hardware, then the installation would be:



1.5.3 Planner Architecture when connecting to multiple Gateways

Other configurations are possible. They will depend on the capabilities of the Host and Planner computer systems, computer architecture requirements and differences. They will also be determined by factors such as total number of Stops to be scheduled, logical divisions that may exist within the scheduling task, the geographic distribution of the Planner installations, and many other such factors.

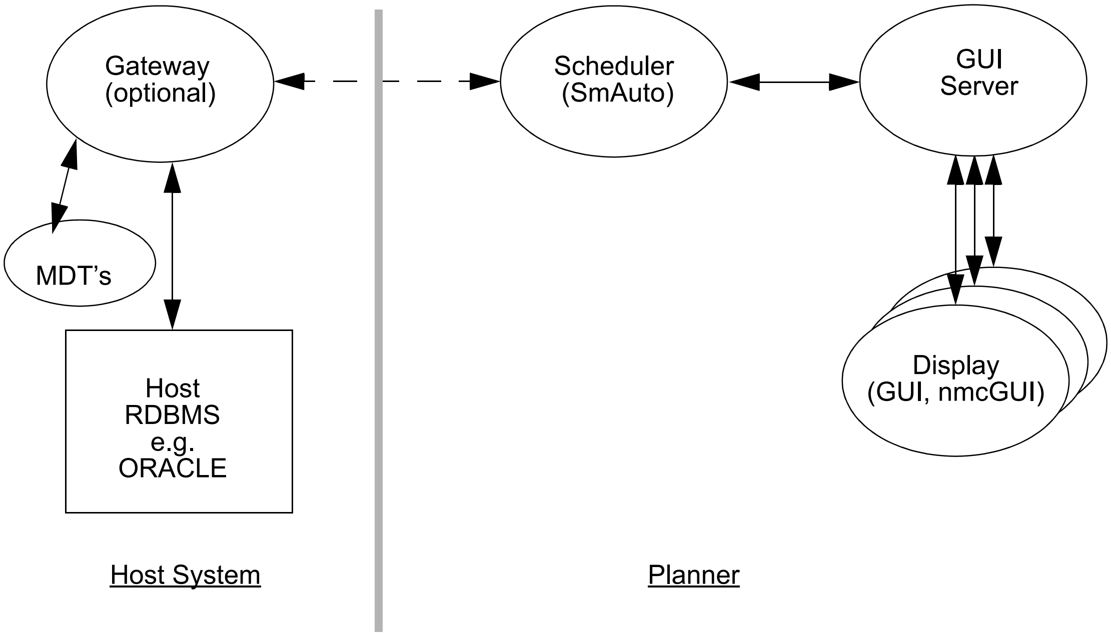


In implementations, where there are several Planner installations being serviced interactively from the same Host system database, it may be preferable to employ multiple Gateway interfaces to ensure that the rate of communication with Planner is not slowed down by the additional message processing that would be required if only one Gateway is used. This would need to be balanced against the additional software maintenance that would be required for the additional Gateway modules and database tables.

Multiple Gateway processes can also be used as a simple way to physically separate two different scheduling tasks for administrative reasons.

1.5.4 Planner Architecture with GUI Server

In situations where multiple PlannerClients require connection to SmAuto, a GUI Server is essential to minimise the network traffic across WAN's and to reduce the load on SmAuto.



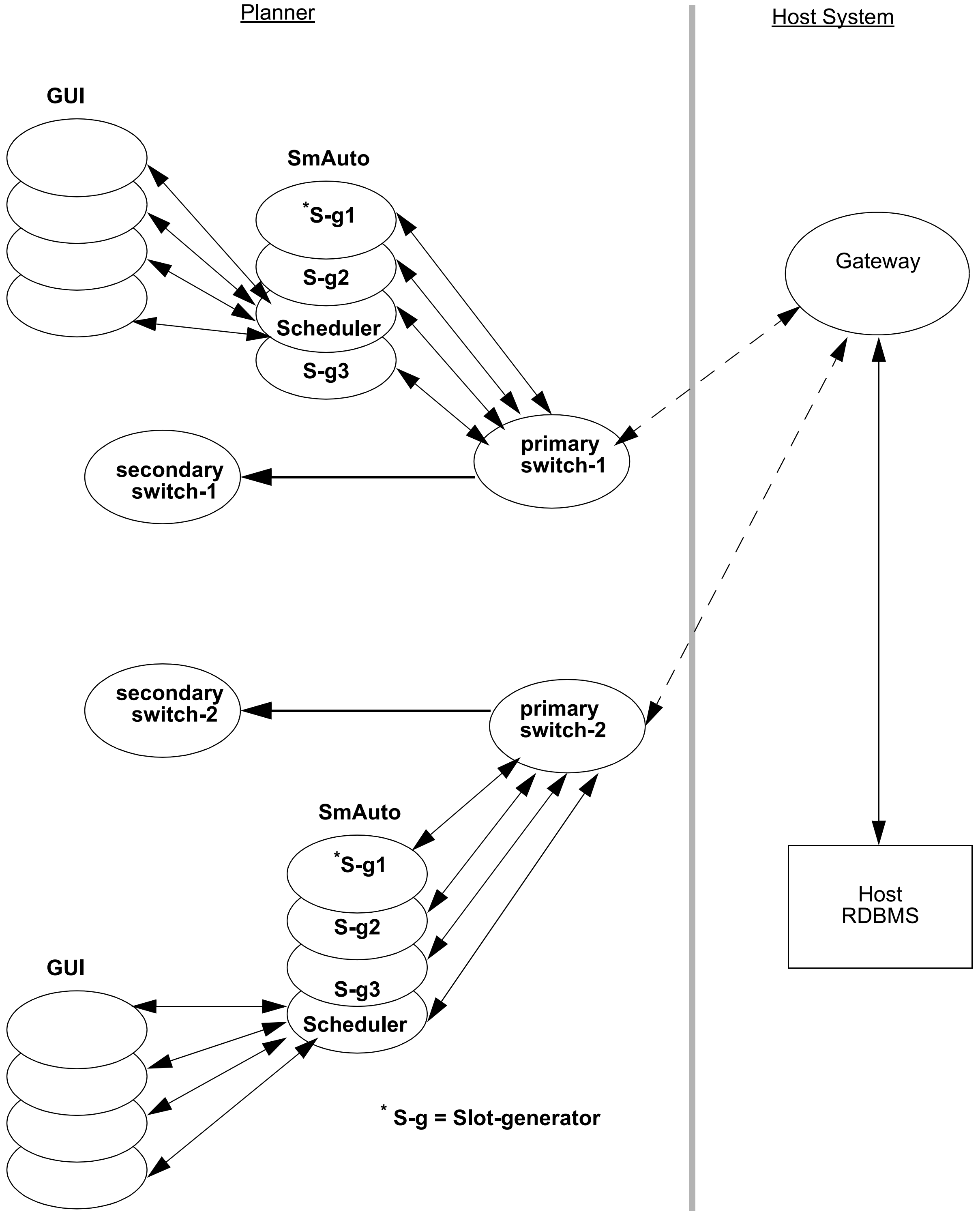
A GUI Server connects to SmAuto as a normal client and also accepts connections from Clients and essentially act as SmAuto. Messages are passed through from Clients to SmAuto and vice versa.

1.5.5 Planner Architecture with Switch

When more than one geographical region is to be scheduled from different installation and hardware, then another Planner process Switch comes into the scene.The Switch is the interface between Planner SmAuto processes on one side and the Host system on the other side. The processes involved in this type of architecture are:

- Scheduler - process name SmAuto - compares the relative (notional) cost of all possible solutions and creates the best solution.
- Slot-generator - process name SmAuto - services slot requests.
- Display - process name GUI or nmcGUI - a Graphical User Interface that handles all screen display, keyboard and mouse activity.
- Switch - process name plannerSwitch - acts as interface between Planner SmAuto process and the host system. The Switch interacts with the host system optionally through a Gateway.
- Gateway - an optional module that interfaces with a Host System Database.

The Switch is the interface between all SmAuto processes and the Gateway.



Chapter 2: Basic Planner Concepts

This chapter describes the data types that can be provided by the Host System to allow scheduling to take place, and how Planner organises the basic data into meaningful arrangements (or schedule solutions).

2.1. DATA TYPES

The properties of all the Planner objects are given by a set of Data Fields which fully describe the particular object. For example, a Shift has a start and finish time, a Stop has a defined location etc.

The Planner objects for which the Host System can provide data are described below, and a full description of all data fields for each object is in Chapter 7: Packets (10.5).

2.1.1 Mandatory or Optional

Because of the ability of Planner to be flexibly configured to suit a particular scheduling task, the data types can be divided into two groups:

- Mandatory Objects - must always be provided by the Host System.
- Optional Objects - may be provided by the Host System to suit the scheduling task.

Planner must always know about the basic properties of the load carriers (Resources, Shifts), the goods to be carried (Stops, Stop Time Windows) and the travel parameters relevant to the geographical region of operation (Map).

The Host System uses the Optional objects to configure the scheduling task according to the nature of the operation. For example, for a courier-like operation, only the basic objects are required; for a more complex continuous distribution operation it may be necessary to provide information about the location and availability of warehouses (Depots, Depot Time Windows), union rules regarding work periods (Breaks) and variations in travel times according to the time of day (Speed Time Windows).

Each of these data objects have data fields defined. Some of these data fields are Mandatory and others are Optional. If the Mandatory data fields are not defined, then an ERROR is reported and the object is not included in the schedule.

2.1.2 For the Load Carrier

2.1.2.1 Resource

A Planner Resource is a combination of vehicle and driver that is expected to be available for work during a specified work period called a Shift. Planner will allocate work to the defined Resources during the work periods defined in one or more Shifts.

Note: Resources are **mandatory** objects.

2.1.2.2 Shift

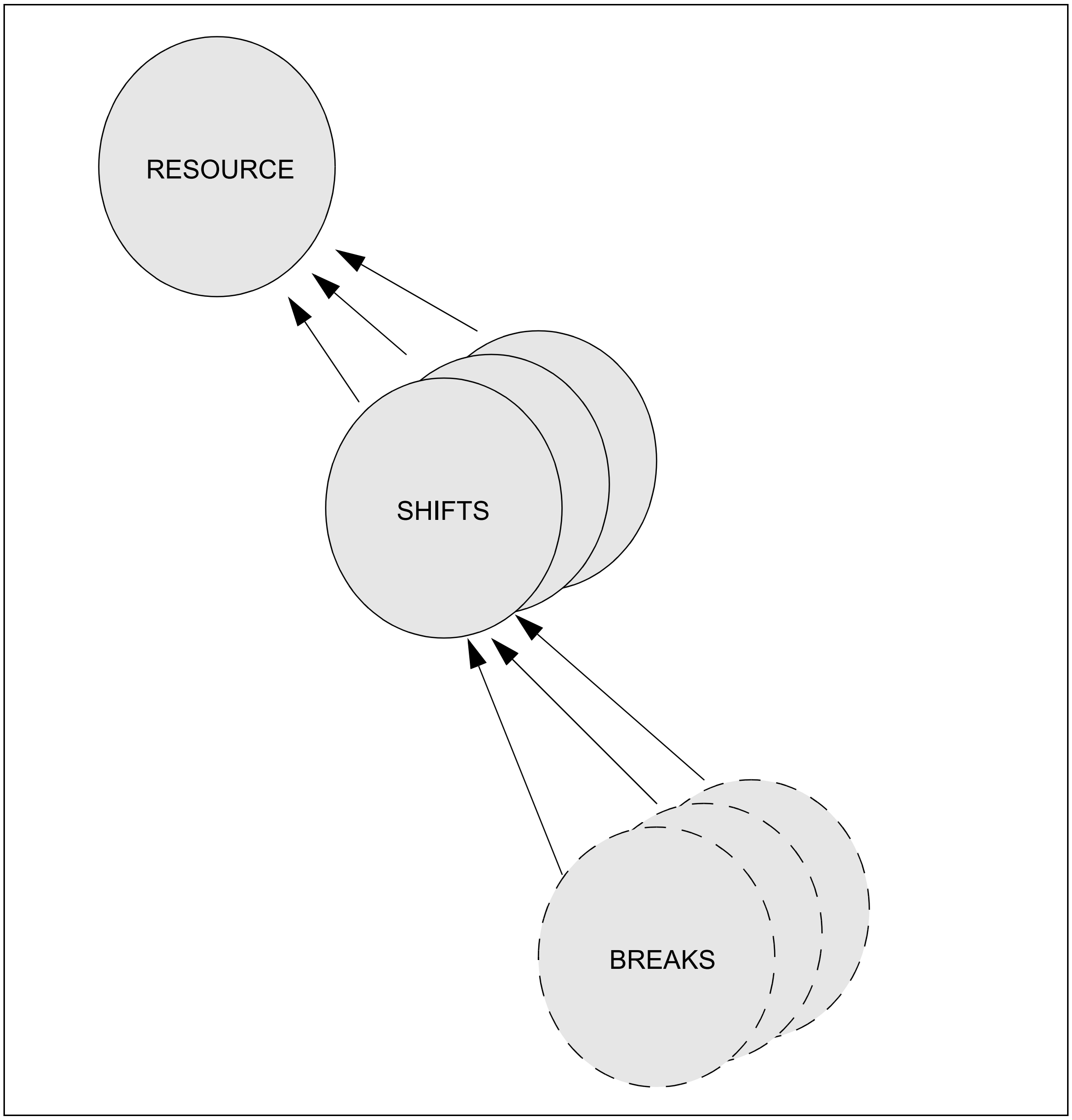
A Planner Shift specifies a set of defining attributes and a period of time. These are used to describe the suitability and the availability of a Resource. A Shift may contain multiple Break periods.

Note: Shifts are **mandatory** objects.

2.1.2.3 Break

A Planner Break is a specified time period in a Shift during which the Shift may not carry out any work or travel between Stops. Breaks can be of different forms, for example, Breaks which are dependent on driving time, elapsed time or time of day.

Note: Breaks are **optional** objects. If no Break data is provided, then Breaks will not be scheduled in the solution



2.1.3 For the Goods being Carried

2.1.3.1 Stops

A Stop is a place to be visited by a Resource. The visit may involve a Collection or Distribution.

Note: Stops are **mandatory** objects.

Stops are expected to be completed during one of the specified Stop Time Windows. Planner will try to schedule the arrival of the Resource to be within a Stop Time Window but, if this cannot be achieved, it will schedule the arrival of the Resource as close as possible to a Stop Time window.

Stops are subdivided into three basic types:

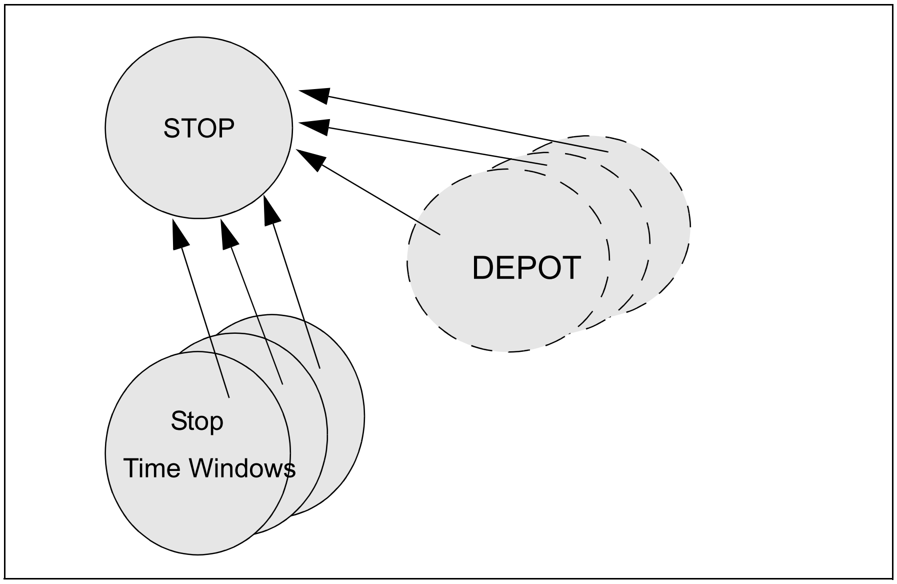
Distribution Stops	- associated with a distribution task from a Depot.
Collection Stops	- associated with a collection task for a Depot.
Stops	- not associated with any Depot.

In addition to the above, there are special types of Stops - Depot, Logon, Logoff, and Break Stops - that are automatically created by Planner when creating a solution. Also see the data types “For Distribution or Collection Tasks” on page 2 - 5.

2.1.3.2 Stop Time Window

A Planner Stop Time Window defines a preferred period of time during which a Resource should visit the specified Stop. There may be several Stop Time Windows associated with a Stop.

Stop Time Windows are **mandatory** objects.



2.1.4 For Distribution or Collection Tasks

2.1.4.1 Depot

A Planner Depot defines a place at which goods will be available or at which goods can be delivered. A Depot has one or more Depot Time Windows during which it will allow access by Resources.

Note: Depot is an **optional** object. If no Depot data is provided, then the scheduling of any Stop and Product data which references a Depot will fail.

Whenever a Resource is required to visit a Depot, a special Stop (called a Depot Stop) will be automatically added by Planner. A distribution stop or a collection stop can be associated with one or more depots.

2.1.4.2 Product Depot

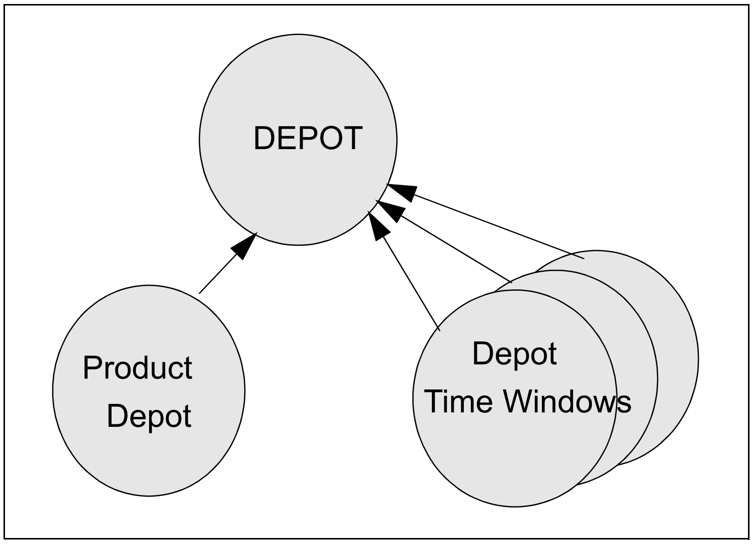
Product Depot defines the product that are available at Depots and product capacity for the depot. It is optional.When Product is specified and linked with the Depot, Depot and Depot Time Windows data becomes mandatory. Without Depot and Depot Time Window, Product data will fail to load.

Any Product can be available from more than one Depot.

2.1.4.3 Depot Time Window

A Planner Depot Time Window specifies a period of access time for Resources at a specified Depot. There may be one or more Depot Time Windows associated with a Depot.

Note: Depot Time Windows are **optional**, but become **mandatory** if Depot data is provided. If no Depot Time Window data is provided, then all defined Depots will be unavailable.



2.1.4.3.1 Depot Time Window Capacity

When delivering goods from or to a Depot, it is often desirable to limit the amount of goods scheduled for delivery from/to the Depot during a given period of time.

Oracle Real-Time Scheduler models this limitation on Depot throughput by assigning a maximum throughput (capacity) to individual Depot TimeWindows. The load transferred to/from a Vehicle during a Depot visit is assigned to one of the Depot's TimeWindows.

When a Depot TimeWindow's assigned load exceeds its capacity, the Scheduler limits further load transfer against this Depot TimeWIndow.

Oracle Real-Time Scheduler selects the Depot TimeWindow for a Depot visit based on the CrewShift within which the visit is scheduled.

This means that all visits by a single Crew within a single Shift will be attributed to a single Depot TimeWIndow, irrespective of how many Depot TimeWindows overlap the Shift.

In order to accommodate the existing business requirements, Planner is designed around the following rules, one of which can be selected through configuration:

- Legacy Planner selects the earliest DepotTW that overlaps the Shift in the appointment Booking request
  Or
  selects the Arrrival Depot Time Window when scheduling
- EarliestOverlappingShift Planner selects the earliest Depot Time Window that overlaps the Shift
- EarliestOfDay Planner selects the earliest Depot Time Window that starts within the Day

2.1.5 For Travel Parameters

2.1.5.1 Geography

The locations of Stops (and the Resources which carry them) are described using latitude and longitude co-ordinates.To accomplish this, Planner uses a “map” containing the geographic position (and names) of common localities.This information is held in map Nodes.

Note: Node data is **mandatory**, and is provided by Oracle Corporation as part of the installation.

Independent Conditional Maps

In conditional maps, the conditions are applied to the map links. Conditional maps can be loaded in conjunction with a standard map.

Each conditional map can have one or more of the following restrictions:

- Vehicle Restriction -Types of vehicle for which map is applicable. By default no vehicle restriction means map is applicable for any type of vehicle.

- Type of Goods Restriction -Types of goods for which map is applicable. By default, no vehicle restriction means map is applicable for any type of vehicle.
- Weight Restriction -Maximum load weight for which map is applicable. By default, no weight restriction map is applicable for any amount of load.

Note: These conditional maps are optional. Sidewinder can develop and install these conditional maps on demand.

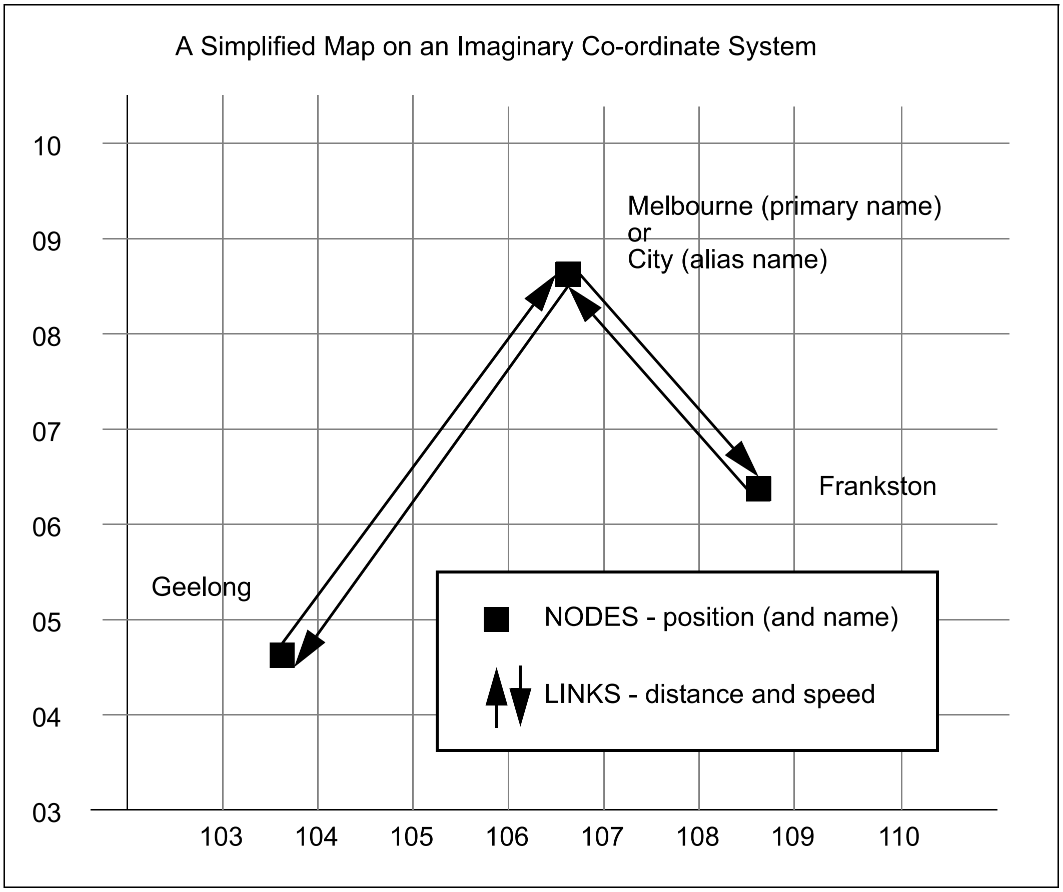
Dynamic Matrices

Dynamic matrix eliminates the need for regenerating map matrices and writing to disk. When a row is required that is not already in memory it will generate it from scratch and store it in the cache. This feature can be set using the “Store Index Table” setting from Planner’s Launchpad. For Launchpad details, please refer the companion Planner Installation manual.

2.1.5.2 Travel

The map contains the time and distance values for a trip between any two Nodes. This travel data is stored in the map as Links, and internally in Planner as a Travel Matrix.

Note: Link data is **mandatory** data. It is provided by Oracle Corporation as part of the installation.


A Simplified Map on an Imaginary Co-ordinate System
10
09
08
07
06
05
04
03
103
104
105
106
107
108
109
110
Melbourne (primary name)
or
City (alias name)
Frankston
Geelong
■ NODES - position (and name)
LINKS - distance and speed

This Map is not a street map, although it works in a similar fashion by providing Planner with a detailed travel path, or route, from one point to another. The travel times contained in the route allow Planner to test for possible violations in Stop execution times and to make good estimations of the location of resources at some future point of time.

When the co-ordinates of a Stop are not exactly coincident with a map Node, the nearest map Link is determined and an additional travel time is factored into the travel calculations.

2.1.5.3 Speed Time Windows

Planner Speed Time Windows specify a period of time during which a specified factor will be used to modify the default travel speeds in the Map. Speed Time windows are used to indicate possible slow downs in a particular area due to traffic congestion, weather conditions or other such factors. In cases where more than one Speed Time Window exists for the duration of a journey, correct Speed Time Windows are applied to the appropriate portion of the journey to give accurate Total travel time.

Note: Speed Time Windows are **optional** objects. If no Speed Time Windows are provided, then the Map speeds will be used unaltered.

2.1.6 For Scheduler Control

2.1.6.1 Cost Control

Planner Cost Item objects consists of a number of notional cost values which will override the default values specified when the software is initially installed.

Note: Cost Item values are **optional** objects. If no Cost Item data is provided, then the default cost parameter values are used.

2.1.7 Data Type Summary

Data Type	Usage	Description
Resource	Mandatory	Contains data describing the load carrier
Shift	Mandatory	Contains data describing the availability and properties of a Resource
Break	Optional	Contains data describing the break periods for a Shift
Stop	Mandatory	Contains data describing the load to be carried
Stop Time Window	Mandatory	Contains data describing the required Collection or Distribution time
Depot	Optional	Contains data describing a distribution or collection point
Product Depot	Optional	Contains data describing the products and available units for a depot
Depot Time Window	Optional, but Mandatory if Depot data is defined	Contains data describing the availability of a Depot
Map	Mandatory	Contains data describing locations and travel parameters
Speed Time Windows	Optional	Contains data describing variations in travel parameters
Cost Item	Optional	Contains data describing optimization parameters

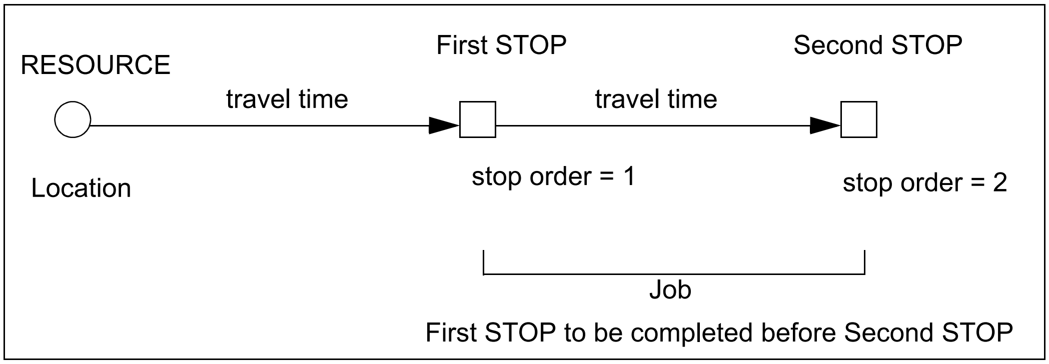
2.2. How Work is Represented

The objective of Planner is to allocate work to Resources in the most efficient manner (either automatically or manually). Planner achieves this by assigning all Stops supplied by the Host to the Shifts in a manner which will adhere to the customer constraints and preferences and results in a low cost, efficient solution.

The arrangement of Stops on Shifts takes into account a number of factors to meet the business requirements of the customers. Planner must take into account the service offered to the customer (e.g., Deliver between 11:00 and 13:00), any restriction imposed on the Stops (e.g., Only use side loading trucks), performance and cost variations for different Resources (e.g., A large truck may be slower and more expensive than a smaller truck). To achieve the best solution all these factors (and more) must be made known to Planner via the data sent from the Host.

A Stop can be defined by the Host as a simple one-stop visit, where the only requirement is that a Resource must visit the Stop to pick up or deliver a load.

In some cases, a number of Stops may be grouped by the Host into a Job. Planner will treat these Stops as a group and will always put all the Stops for the Job on the same Resource, in the same Haul, and in the sequence specified.



This two-stop Job is just one of the many ways in which Jobs can be configured. Jobs may consist of more than two Stops to be completed in fixed or flexible order, and a mix of Stops can be assigned to a particular Shift.

Whenever a Stop requires its Resource to visit a Depot, a special Stop (called a Depot Stop) will be automatically added by Planner onto the Resource. These Depot Stops may represent a Collection or a Distribution, but not both. Planner will arrange the Depot Stop(s) on the Resource to suit the operation, that is, if the Depot is used for the distribution of goods, then the Depot Stop will be arranged on the Resource to occur before the associated Distribution Stop (and vice versa).

Planner must also be informed about any time constraints to be applied to individual Stops, and any other conditions to be applied to the allocation of Stops to Shifts.

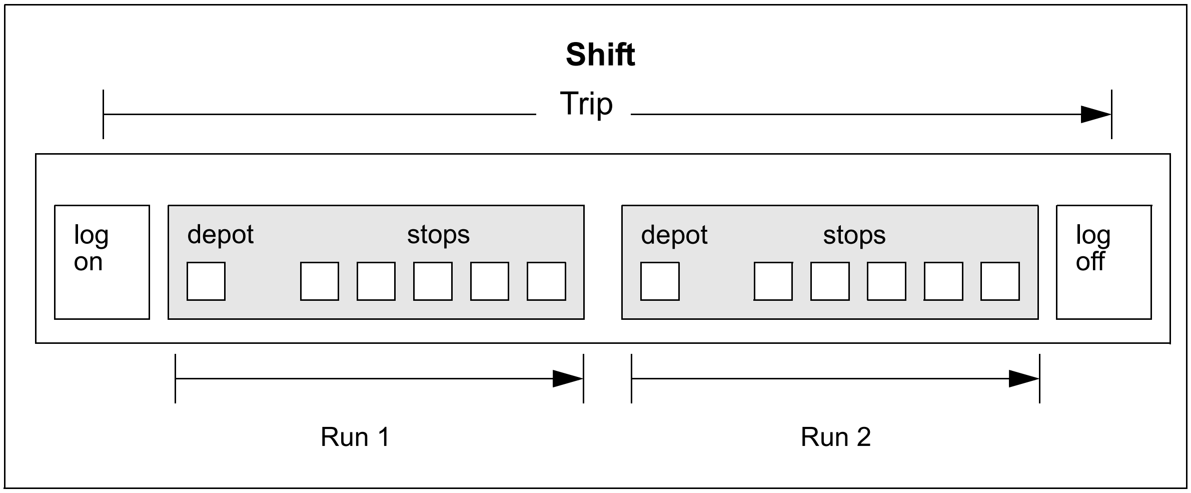
When all the operating rules and constraints on Resources, Stops and Shifts have been made known, Planner will arrange the Stops on the Resources in the most efficient manner.

The allocation of stops is represented as follows:

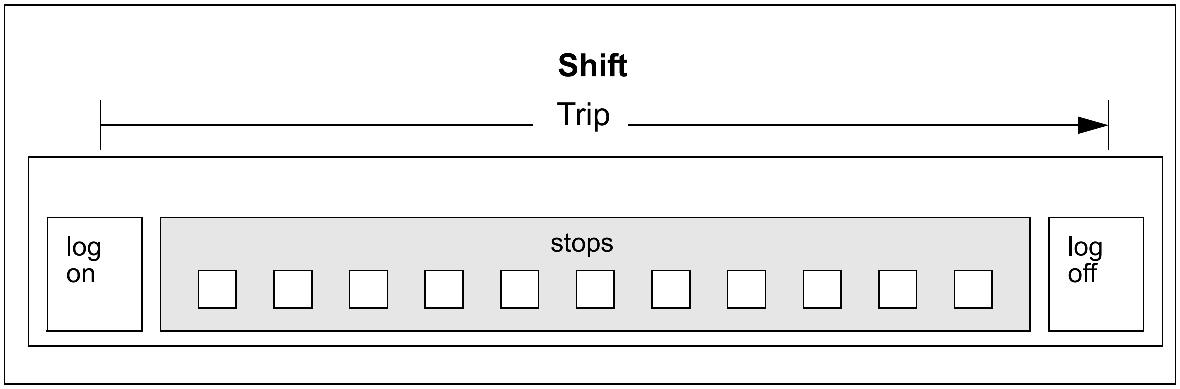
Trip	- the Resource’s activity between Logon and Logoff (that is, a driver Shift)
Run	- an outwards and return journey, usually from or to a Depot.

The Shift designed for a Resource can contain as many Runs as are necessary to complete the required work. The optimum number of runs required will be determined by Planner during the scheduling process.

The daily work load of a Resource, that is, a Shift, doing deliveries from a Warehouse can be represented as follows:



For a courier type operation, the work (shown below) is continuous.:



The following chapters will describe how to set up the various data structures to represent the type of operation which is to be scheduled.

2.3. Multiple Time Zone Handling in Planner

Oracle Realtime Scheduler operates, in Realtime, over a large geographic area that covers multiple time zones.

It allows Object and event times to be entered and viewed in the objects local time as well as in the User’s local viewing time.

2.3.1 Passing Date/Times to SmAuto

SmAuto always deals with absolute times, there are no abstract times used. As such all times passed to SmAuto have an optional UTC offset. SmAuto remembers this offset and includes it when exporting times.

If the offset is not supplied in the input time then the local time zone for SmAuto will be used. This ensures backwards compatibility with previous releases of Oracle Realtime Scheduler.

To generalise the UTC influence on local times, we will be using the following scheme:

```
(HHMM)Z+(HHMM);
which is the same as (Local Time)Z+UTC Offset
```

For example:

0800Z+1000 represents 8am local time which is offset from UTC by 10 hours.

2.3.2 Generating Date/Times within SmAuto

SmAuto generates several date/times during scheduling. When an actual event time, i.e. Shift Start, is passed to SmAuto, the UTC offset information is also included. SmAuto remembers this offset, overrides any previous settings and includes this offset when exporting the event date/time.

2.3.3 Planned Shift Start and Complete Times

The planned Shift Start and Complete times generated by SmAuto includes the same UTC offset as passed to SmAuto for the start of the Shift's Time Window.

2.3.4 Planned Stop Arrival, Start and Complete Times

The planned Stop Arrival, Start and Complete times generated by SmAuto includes the same UTC offset as passed to SmAuto for the start of the Shift the Stop is currently associated with.

Chapter 3: Matching the Operation

This chapter describes how to set up the data structures to represent particular operational requirements.

It is important to remember that the quality of the Schedule solution created by Planner will depend almost entirely on the quality of the data which it receives.

3.1. Operational Imperatives

Before using Planner, it is essential to have a thorough understanding of the nature of the transport operation and the scheduling result that is being requested of Planner.

3.1.1 Scheduling Requirements

Planner can be used to schedule yesterday’s data, today’s data or next year’s data with equal ease. Planner has no concept of the actual time of day unless it is set to work in real time mode. It is more concerned with the amount of time that has elapsed since the specified starting time for the data to be scheduled.

Planner can be used for any one of the following purposes:

3.1.1.1 Batch Planning

Planner is most often used to schedule data for tomorrow or some time in the immediate future. When the schedule is complete, the manifests can be printed and handed over to the drivers.

3.1.1.2 Real Time Planning

When Planner is used to schedule today’s data, it actually runs in parallel with “real” time. When this occurs, Planner can properly interpret real time events such as new or altered Stop data, radio notification of completed Stops and GPS positioning (i.e if the installation is licensed for real time operation).

In this mode, the Planner scheduling task is completed only when all the Stops being scheduled have actually been completed.

3.1.1.3 Performance Analysis

Sometimes Planner will be used to schedule a data set that occurred some time in the past. To do this, the data must be set up so that the dates and times specified for the Resources, Shifts, Stops etc, are properly aligned with each other, rather than being aligned with the current time and date at which the scheduling task is actually taking place.

Scheduling a past data set is often the best way to “Tune” the performance of Planner. The same data can be scheduled repeatedly under different conditions so that the results can be compared with actual fleet performance. It can be used to achieve particular goals, such as varying the “cost” parameters to minimize the amount of late time in the solution.

3.1.1.4 Strategic Planning

Planner can be used to perform strategic planning for the Allocator to answer some operational questions about the size of the fleet or the location and capacity of Collection and Distribution Depots.

For this purpose, a relatively recent data set can be used, but the dates and time used in the data set are not critical. The important thing here is that same data set is scheduled repeatedly, each time with a different values for the items of interest. For example, a different Depot location might be specified for each schedule operation.

3.1.2 Operational Requirements

The only way to decide the right data set to be used with Planner is to examine the transport operation itself and identify the crucial and important elements.

It is important to identify facts such as whether the operations performed are primarily:

Distribution from Depots?

Collection to Depots?

Random Collections and Distributions?

Longhaul operations?

Continuous operations?

A combination of a few or all the above?

It is also important to consider things such as:

Does the fleet always start and stop at the same location?

Are all vehicles the same or do they differ in size, volume and other such features?

Do the drivers take mandatory breaks?

How important is overtime?

How important is customer service?

How flexible is the time for Collections and Distributions?

All of these operational requirements, and more, can be made available to Planner so that the scheduling solution reflects the business requirements of the operation as closely as possible. The remaining part of this chapter describes the elements that make up the data set used for scheduling, and gives some examples of how they are used in typical transport operations. Also see “Setting Up the Data” on page 3 - 4, and “Operational Scenarios” on page 3 - 20.

The data definitions for Planner are given in Appendices. By observing the Data Packet structure and the format of the data fields contained in them, it is relatively easy to match the operational requirements that are important to the Data Packet fields that are passed to Planner.

It is essential that the Host system has immediate access to the necessary operational data and that the requirements are specified in such a way that the data can be used to construct data files or message packets that can be used in Planner.

It is the responsibility of the Host system to make sure that the required data is accurate and sent in the proper format to Planner. At the very least, the mandatory fields of the Data Packets must be filled in. Otherwise the Data Packets will be rejected or ignored by Planner. The optional fields in the Data Packets can be used to override the standard settings of Planner so as to reflect the operational requirements of the installation as accurately as possible.

3.2. Setting Up the Data

The Planner data can be set up to reflect a wide variety of operational requirements.

3.2.1 For the Load Carrier

Resource is the object that identifies the basic load carrying mobile in Planner. It is an mandatory object. It is basically a combination of vehicle and driver, both of which should be available during a specified work period called a Shift.

A Resource can be associated with several different Shifts to simulate a continuous operating environment, if required.

The capabilities of the Resource can be defined in the Shift data. They are assumed to exist only for the duration of the Shift even though some of them (e.g.,MAX_WEIGHT, MAX_VOL) are fixed and some of them are transient (such as the capabilities of the driver, e.g., ATTRIBUTES).

Each Shift can have multiple idle periods defined (Breaks) that will automatically be scheduled into the solution.

One or more Shifts can be grouped together into Hauls to properly represent different operations. Strictly speaking, a single Shift is a one-Shift Haul but it is still referred to as a Shift. Several Shifts that are grouped together are called Hauls, and they can be grouped into Manyshifts or Multishifts (for continuous operation) or Longhauls for tasks that cannot be completed in a single Shift. Also see the section on “Creating Hauls” on page 3 - 8.

Note: The Shift group (Haul) must exist in the data set as well — the Planner cannot create Shift groups, although the Planner can alter the nature of the Haul.

3.2.1.1 Resource

The Resource Data Packet carries mandatory information that uniquely identifies the Resource and specifies its location (place and time) at the start of scheduling. The specified location is usually the place at which the vehicle is garaged. This may be different from the location at which the Resource is expected to Log-on. This information is used for travel time calculations.

3.2.1.1.1 Resource Cost

The instantiating cost of the Resource can be modified. The instantiating cost (RSRC_COST) can be thought of as the standing cost associated with the use of this Resource. This cost can be used to determine how frequently the resource is used while scheduling.

3.2.1.2 Shift

The Shift Data Packet carries mandatory information that uniquely identifies the Shift, and defines a work period for the associated Resource. There must be a separate Shift Data Packet for every work period definition — Shifts cannot be used multiple times.

3.2.1.2.1 Shift Types

The mandatory data field TYPE defines the nature of the operation being carried out. This field defines continuous, discontinuous or longhaul operations.This, in turn determines the conditions under which the Resource is expected to start and finish the Shift. The Shift types defined are:

STANDARD - The Shift commences at the Log-on position and terminates at the Log-off position. No goods may remain on the Resource at the Log-off position.

ONE_WAY - The Shift commences at the Log-on position and terminates at the Log-off position. But the total cost calculation for the shift does not include the travel distance and time between the last stop and log-off stop. No goods may remain on the Resource after it finishes at the Last Stop.

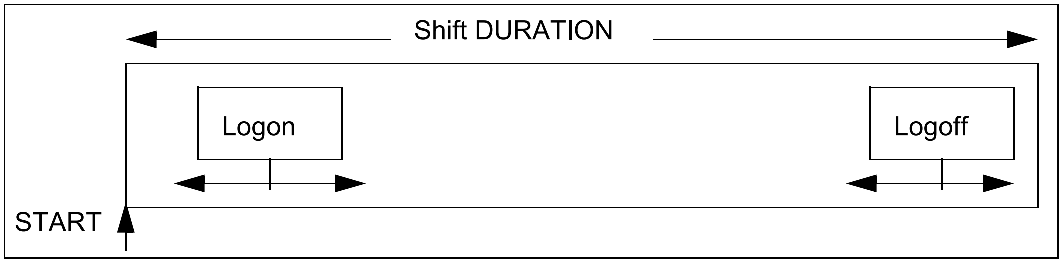
MANYSHIFT - The Shift commences at the Log-on position and terminates at the Log-off position similar to a STANDARD shift, but the starting time of the next Shift depends on the finishing time of the current shift. No goods remain on the Resource at the Log-off position.

MULTISHIFT - This Shift is similar to a MANYSHIFT. Each Shift commences at the Log-on position and terminates at the Log-off position. The start time of the next Shift depends on the finish time of the current shift. Goods may be carried from one shift to the next but not beyond the final shift. The final shift has to be of the type Standard.

LONGHAUL - The first shift starts at its Log-on position, the final Shift terminates at it’s Log-off position. Intermediate shifts do not necessarily start or finish at the designated position but will “carry forward” the previous log-off position into the next shift. Goods may be carried into the next but not beyond the final shift. The final shift has to be of the type Standard.

3.2.1.2.2 Shift Duration

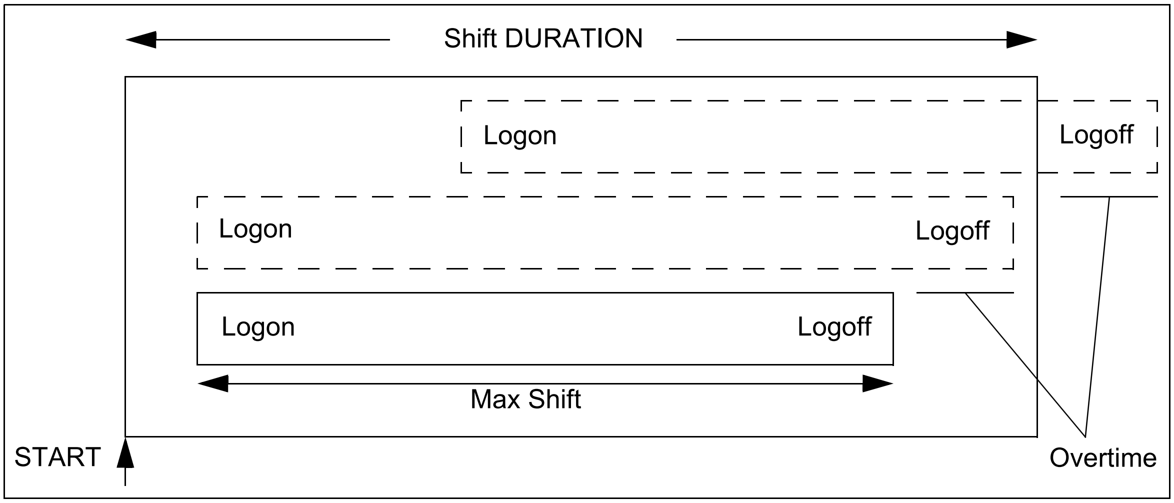
The mandatory START_DATE, START_TIME and DURATION fields define a time window within which the Shift should be started and finished. This gives Planner great flexibility to optimize the Log-on and Log-off times of the Resource for maximum efficiency.



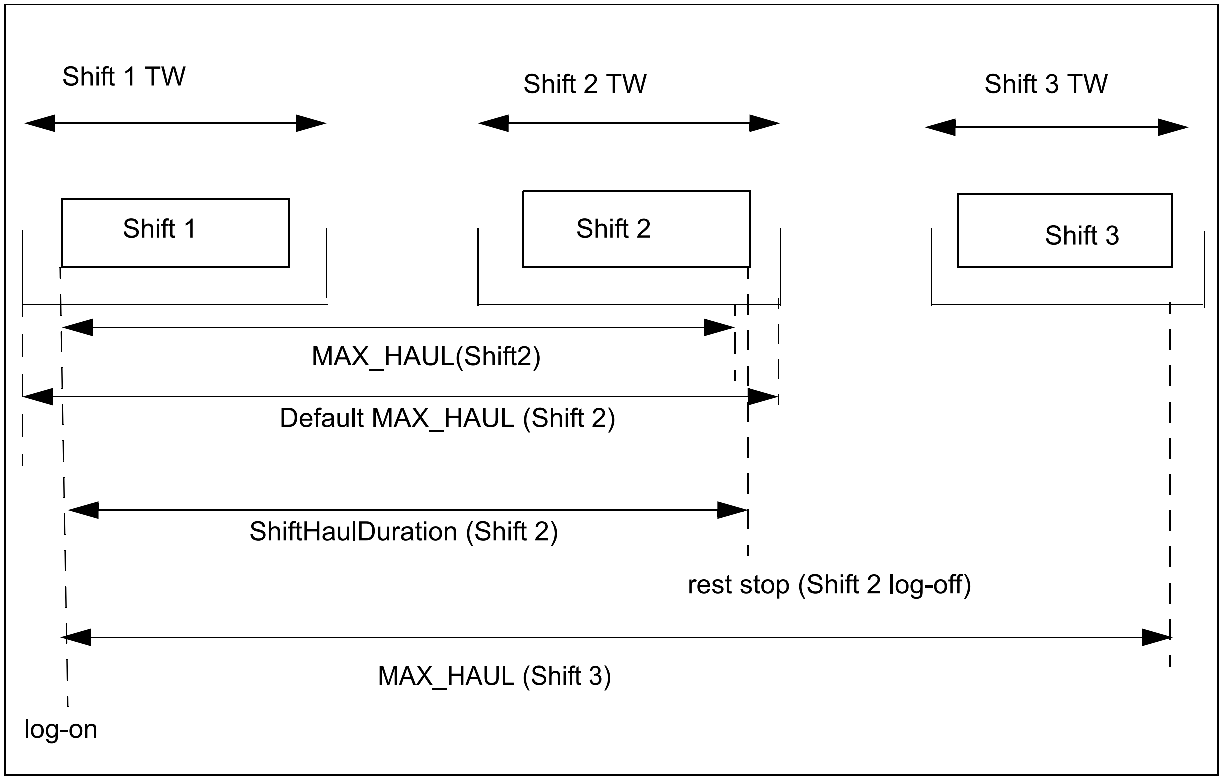
3.2.1.2.3 Shift Working Hours

The MAX_SHIFT field can be used to define the maximum time period of work for the Resource within the time window. If this field is not set, the working hours for the Shift are assumed to be the same as the duration of the Shift.

Overtime, that is, work outside either the MAX_SHIFT or Shift DURATION, can be more or less discouraged by changing the OVERTIME_COST field of the Shift relative to other shifts. If this field is not set, then overtime will not be discouraged.



MAX_HAUL can be defined as the maximum length of a haul at the end of the corresponding shift and can be specified for each shift of a haul.



In addition to the existing MAX_SHIFT parameter, MAX_HAUL can be considered as a generalization parameter to LongHaul, MultiShift and ManyShift, and a further improvement of Scheduler functionality. The MAX_HAUL and MAX_SHIFT parameters are independent. If MAX_HAUL for a shift has not been defined then the default value of MAX_HAUL is a time interval between the start time of the first shift time window and the end of the current shift time window. HAUL_EXTEND_COST will be applied to a shift if MAX_HAUL value is less than the corresponding ShiftHaulDuration. See HAUL_EXTEND_COST on page 45.

For the specific case of a longhaul Shift, Multishift and Mannyshift, the Shift data also specifies the minimum REST_PERIOD that must follow the Shift. If this field is not set, then no rest period will be scheduled into the solution.

3.2.1.2.4 Shift Cost

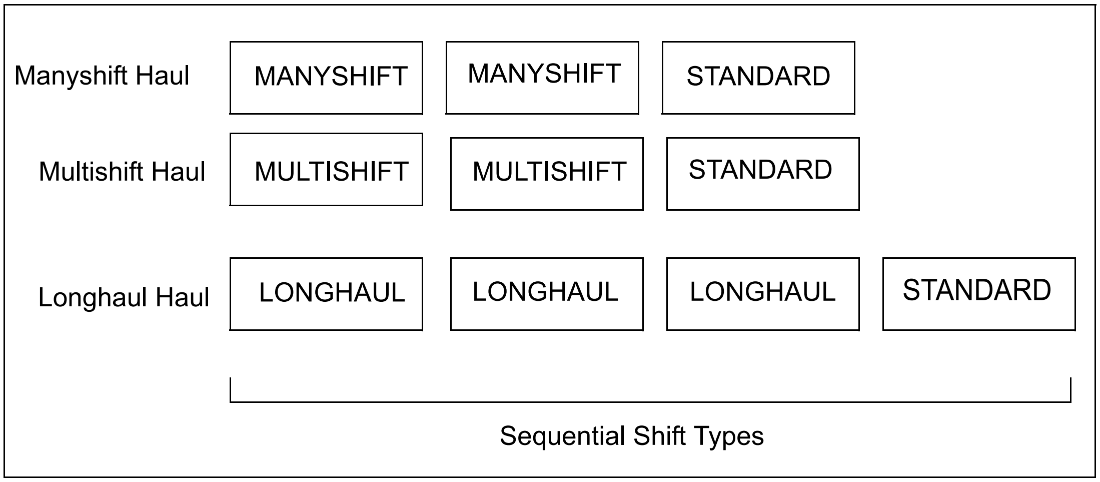
The Shift data carries some fields that can be used to alter the instantiating and running costs (SHIFT_COST, DIST_COST, TIME_COST) of this Shift relative to the other Shifts known to Planner. Variations in these costs may be used to encourage or discourage the use of this particular Shift — things such as discouraging the use of contractors or encouraging the use of more efficient vehicles first.

If these values are not set, then no special preference will be given to this Shift.

There is also a LONGHAUL_COST which is applicable only when the shift is of TYPE LONGHAUL. It is a relative cost used to either encourage or discourage the driver from taking a REST_PERIOD at the designated logoff position.

3.2.1.2.5 Creating Hauls

Hauls are set up in the data set by creating the required number of sequential Shifts of the same type for the same Resource, where the terminal Shift in the Haul must be a STANDARD type Shift.



Also see“Single / Multiple Day Scheduling” on page 3 - 20.

3.2.1.2.6 Load Limits

Resource capacity can be specified in the MAXWEIGHT and/or MAXVOLUME fields. If these fields are not specified, the Resource is a bottomless pit.

The units by which loads and capacities are described must be used consistently in all data specifications.

3.2.1.2.7 Parcel Size

The value MAXSIZE specifies the maximum “size” of any individual parcels (partial load) that can reasonably be handled by the Resource. For example, it might be set to the maximum weight that a driver can carry. The Shift will not be considered for Stops that have a size specified that is greater than MAX_SIZE.

If MAX_SIZE is not set, then it is assumed that all parcels irrespective of parcel size can be properly handled. The units by which loads and capacities are described must be used consistently in all data specification.

3.2.1.2.8 Zones

Zone preferences can be used to specify a preferred geographical region for work. Zones may be used to reflect a driver's familiarity with a particular region or maybe just a radio transmission area. The set of zone names used is arbitrary, but must be the same as the set of zone names used when defining the Stop data.

If a zone name is specified in the ZONE_MUST field of the Shift, then the Shift can only be assigned Stops that have the same zone name specified.

If a zone name is specified in the ZONE_PREF field of the Shift, then the Shift may be assigned Stops outside the specified zone, but only if the solution is efficient.

If no zones are specified, there is no constraint. Multiple zones are allowed.

3.2.1.2.9 Attributes

Attributes describe the permanent skills or properties possessed by the Resource, and can be used to make sure that the Resource has the right "tools", such as a Tail Lift or Union Ticket, when it arrives at the customer location. The set of Resource attribute names used is arbitrary, but must be the same as the set of Resource attribute names used when defining the Stop data.

If no Resource attributes are specified, then none are assumed to be possessed by the Resource.

Note: Not to be confused with acquired Job attributes which are temporarily transferred to the Resource from a Stop while the load is being transported.

3.2.1.2.10 Relative Speed

The travel speed of the Resource, relative to the travel speed specified in the map, can be modified in the RELSPEED field to reflect a slower or faster vehicle, or a driver with more or less experience. The travel speed of the Resource is used to compute the arrival times at the Stops, so RELSPEED should be set to a value that properly describes Resource travel.

If RELSPEED is not set, then normal travel times (calculated internally in the map travel time matrix) are assumed.

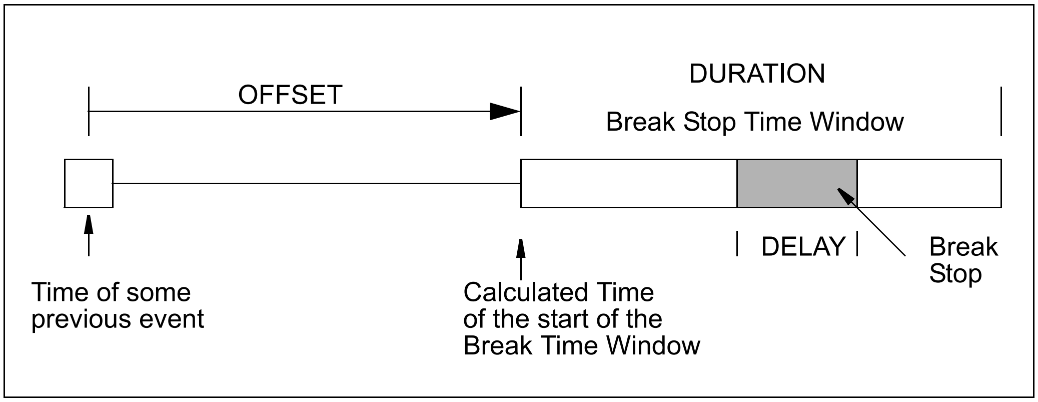
3.2.1.2.11 MDT (Mobile Data Terminal) messages

The MDT field is an optional field where a unique identifier for the MDT can be specified. It uniquely identifies the MDT associated with the shift. It enables communication between the Scheduler and the driver in the vehicle carrying the MDT terminal.

3.2.1.3 Break

The Break Data Packet is an optional object. The Break Identifier is a mandatory field that uniquely identifies the Break. It defines the idle period of rest time associated with the shift. There must be a separate Break Data Packet for every rest period defined — Breaks cannot be used multiple times.

The mandatory Break data field DELAY defines the length of the rest period, while the mandatory fields TYPE, OFFSET and DURATION are used by Planner to calculate a Time Window within which the Break must commence.


DURATION
OFFSET
Break Stop Time Window
DELAY
Break Stop
Time of some previous event
Calculated Time of the start of the Break Time Window

The TYPE field specifies the previous event from which Break timing is calculated. For the various Break types, the specified OFFSET time is interpreted as:

SHIFT	- elapsed time from the earliest possible start of the Shift.
LOGON	- elapsed time from the predicted or actual time of Logon.
ELAPSED	- elapsed time from the predicted or actual start of the previous Break or Logon.
DRIVER	- accumulated driving time from the predicted or actual start of the previous Break or Logon.

The calculated Time Window is used by Planner to position the Break Stop at the most efficient position in the Shift.

3.2.1.3.1 Break Order

If a Shift is to contain multiple Breaks, then the BREAK_ORDER field must be set in each defined Break to specify their order of execution. If this field is not set then all the Breaks defined are assigned a default BREAK_ORDER value of 1 and the order of execution of multiple Breaks in a Shift will be decided by the order in which they are received by Planner.

3.2.2 For Distribution or Collection Tasks

For operations that include the Distribution of loads from a Depot or the Collection of loads for return to a Depot, its is necessary to make the details of each available Depot known to Planner.

Each Depot can have several access periods defined (Depot Time Windows) that are used to control the period of availability of the depot for Collections or Distributions.

3.2.2.1 Depot

The Depot Identifier is a mandatory field that uniquely identifies the Depot, and defines its exact location and the nature of its operation, that is, Distribution or Collection.

3.2.2.1.1 Depot Delays

The Depot Data Packet can specify a time delay (SITE_DELAY) when entering or leaving the Depot. All stops visiting the Depot may be delayed by a certain amount of time (SITE_DELAY) involving delays due to security check, signing in etc.

Note: Any time delay associated with loading or unloading individual loads is specified in the Stop Data Packet for the load (DEPOT_DELAY).

The total delay at the depot is calculated as a sum of the SITE_DELAY plus the DEPOT_DELAY values specified for each of the stops visiting the Depot.

Also a relative speed of travel in the vicinity of the Depot (RELSPEED) can be specified so that the speed of the vehicle (and hence the time taken by it) can be calculated accordingly.

3.2.2.2 Product Depot

The PRODUCT_ID is a mandatory field that identifies the product and binds it to a depot using the DEPOT_ID field that uniquely identifies a product depot. MAX_UNITS field is used to set the maximum unit of the product that can be held at the depot. TARGET sets the units of the product that can be collected at the depot.

3.2.2.2.1 Product Depot Cost

PRODUCT_COST is used to encourage or discourage the product being collected at the referenced Depot. TARGET_PRIORITY is used to obtain the most cost effective product depot relationship towards meeting the product target, whenever more than one depot is capable of collecting the product.

3.2.2.3 Depot Time Window

The Depot Time Window Data Packet carries information that uniquely identifies the Depot Time Window, specifies the Depot with which it is associated, and defines a Resource access period. There must be a separate Depot Time Window Data Packet for every access definition. Depot Time Windows cannot be used multiple times for access to the same resource.

3.2.2.3.1 Time Window Costs

The arrival of a Resource outside the specified Depot Time Window interval can be discouraged more or less by changing the LATE_COST data field value.

Where several Depot Time Windows are associated with the same Depot, the preference for each window can be set using the WINDOW_COST data field.

If these fields are not set, no special preference will be given to this Depot Time Window.

3.2.3 For Loads

The basic load object which must be defined for Planner is known as a Stop. For Collection or Distribution operations, Stops can be associated with a particular Depot or with any number of specified Depots. Stops can be grouped into Jobs for transportation purposes, where in all the Stops associated with the Job must be assigned to the same Resource.

Each Stop can have several access periods defined (Access Time Windows) that are used to control the Resource access time period to the Stop. In addition to Access Time Windows, Stops can also have multiple Service Level Agreement (SLA) Time Windows.

3.2.3.1 Stops

The Stop Data Packet carries mandatory information that uniquely identifies the Stop, specifies its exact position and defines the nature of the operation at the Stop.

If the Stop is of TYPE PKUP or DROP, then it must have one or more Depots associated with it. Type PKUP or DROP imply that the Stop is part of a Collection or Distribution operation.

If the Stop is of TYPE PTP (“Point to Point”) or POU ("Period Of Unavailablity"), then the Stop has no Depot associated with it. PTP Stops can exist either as a single stop job or as a job with two or more stops in it. Jobs which involve only visits as part of a service requirement and hence has no goods transaction results in single stop PTP jobs. On the other hand the more popular PTP scenario involves two or more stops with at least one stop involving a Collection and the other stop involving a Distribution. "Period Of Unavailability" associated with a Shift is represented as a Stop and is scheduled as a normal Stop.

Note: Other stops required to make up the work for a Shift (that is, Logon, Logoff, Break and Depot Stops) will be created automatically by Planner as and when required in the schedule.

3.2.3.1.1 MDT identification

The fields JOB_CODE and STOP_CODE are optional fields which are used to uniquely specify the Job details and the Stop details to the driver/resource with a MDT. When the Allocator despatches a job to the driver from the Planner^{TM} Client or just sends in a plain text message, the JOB_CODE and STOP_CODE can be specified. These fields tell the driver which Stop to go to. It also tells the driver whether this particular task is part of a Job group. The driver can send back messages reporting the status of the task specifying the STOP_CODE and optionally the JOB_CODE to the Scheduler.

3.2.3.1.2 Load Weight and Volume

The WEIGHT and/or VOLUME fields can be used to specify the amount of load to be carried. These values are used to make sure that the Resource is not overloaded. If the Resource capacity is specified in both MAX_WEIGHT and MAX_VOLUME, then both WEIGHT and VOLUME should be specified for each Stop. If they are not set, then the load is assumed to be zero. The units by which loads and capacities are described must be used consistently in all data specification.

If the Stop is a Collection Stop, then the amount of load being picked up must be specified as a positive number. If the Stop is a Distribution Stop, then the amount of load being delivered must be specified as a negative number.

Where several PTP Stops are combined to form a Job, then the summation of all the loads in the Job must be zero. The Collection Stops must have lower JOB_STOP_ORDER values than the Distribution Stops (to make sure that the Collections come before the Distributions).

3.2.3.1.3 Parcel Size

The SIZE field specifies the “size” of the largest parcel (partial load) at the Stop. It is used to ensure that goods are handled properly. A Stop will not be assigned to a Shift if it’s SIZE is greater than the MAX_SIZE of the resource used in that Shift.

If SIZE is not specified, then the parcels are assumed to have no “size”.

3.2.3.1.4 Delays

The relative travel speed in the vicinity of the Stop location can be altered using the RELSPEED field. This field is used where travel near the Stop such as in an industrial complex, differs greatly from normal road speeds. If RELSPEED is not set, then normal travel speed is assumed.

The JOB_DELAY field can be used to specify the amount of time taken to load or unload the Resource at the Stop location.

If the Stop is part of a Collection or Distribution operation, then the DEPOT_DELAY field is used to describe the time taken to load or unload the amount specified for this Stop at the Depot.

These delays are used to estimate elapsed time. If these values are not set, then no delays will be anticipated at the Stop location.

3.2.3.1.5 Site Delays

A specified Stop location which is used for more than one related or unrelated Stops by the same or different resources may be described as a SITE. The judicious use of sites is advantageous to Planner because it can minimize the number of visits by different Resources to the same local area. The SITE itself can cause further time delays that can be specified in the SITE_DELAY field.

This delay is used to estimate elapsed time at the Site.

3.2.3.1.6 Site Separation

In certain industrial establishments, it is quite common to come across a situation where a Site needs to be visited repeatedly. These visits to the same site can occur only after a certain interval of time. SITE_SEPARATION_ID identifies the Site to which visits need to be regulated and the SITE_SEPARATION_TIME identifies the delay between each visit to the Site.

3.2.3.1.7 Resource Attributes

The Stop data can specify several sets of Resource attributes that will be used to encourage, discourage or prohibit certain Resources being used for this Stop. Planner compares the Resource attributes specified in the Stop with the permanent attributes possessed by the Resource to find the most appropriate shift. The attribute fields are:

R_MUSTHAVE - Resources with these attributes are included.
R_MUSTNOTHAVE - Resources with these attributes are excluded.
R_PREFHAVE - Resources with these attributes are encouraged.
R_PREFNOTHAVE - Resources with these attributes are discouraged.

For example, a Stop consisting of a large heavy load may insist on a Resource with a “Tail Lift” in the R_MUSTHAVE field, or a Stop at a Bank may insist on a driver without a “Helmet” in the R_MUSTNOTHAVE field.

When Planner is considering Resources for this Stop, it may sometimes assign a Resource that violates the “preferred” Resource attribute requirements if the solution is efficient.

If no Resource attributes are specified, then all Resources available will be considered for the Stop.

3.2.3.1.8 Transferable Job Attributes

There are some attributes specified in the Stop that can be “transferred” to the Resource for the time that the load is being transported. These temporary Job attributes are specified in the J_PKUP and J_DROP fields.

For example, if the Stop involves the transportation of reactive chemicals, then the J_PKUP and J_DROP fields could be set to “Hazardous”. The Resource would temporarily acquire that Job attribute, so that Planner would not allocate any other inappropriate work to that Resource.

If these fields are not set, then no attributes will be transferred to Resources.

Where several PTP Stops are combined to form a Job, then the summation of all the transferable attributes in the Job must be zero.

3.2.3.1.9 Job Attributes

The Stop data can specify several sets of Job attributes that will be used to encourage, discourage or prohibit certain Resources being used for this Stop. Planner compares the Job attributes specified in the Stop with the temporary Job attributes possessed by the Resource to find the most appropriate Resource. The attribute fields are:

J_MUSTHAVE - Resources acquiring these attributes are included.
J_MUSTNOTHAVE - Resources acquiring these attributes are excluded.
J_PREFHAVE - Resources acquiring these attributes are encouraged.

J_PREFNOTHAVE - Resources acquiring these attributes are discouraged.

For example, if a Stop involves the transportation of combustible material, then the Stop may wish to exclude Resources that are already carrying hazardous goods. In this case, the Stop would set "Hazardous" in the J_MUSTNOTHAVE attribute field.

When Planner is considering Resources for this Stop, it may sometimes assign a Resource that violates the "preferred" Job attribute requirements if the solution is efficient.

If no attributes are set, then all Resources will be considered for the Stop.

3.2.3.1.10 Exclusivity

A Stop can specify that it requires exclusive use of a Resource at the time it is picked up by setting the SINGLE_ITEM field. If not set, no exclusivity will be scheduled. If SINGLE_ITEM is set to YES, then the resource must have no other load on board when visiting this stop.

Note: This field can be used in conjunction with the SINGLE_LINK field to ensure that this Stop has exclusive use of the Resource until it is delivered.

SINGLE_LINK can be set to ensure that the Stop/Stops comprising the job must be visited in a row without other stops being visited in between. It is used to encourage completion of a sequence of stops as quickly as possible.

3.2.3.1.11 Zones

If required for dispatching purposes, the ZONES field can be used to specify the zone(s) to which this Stop belongs. This information is used by Planner to prohibit or discourage the assignment of the Stop to Resources that do not work in the specified zones. See "Zones" on page 3 - 8 for related Shift zone details.

If the ZONES field of the Stop is set to AUTO, then Planner will automatically set the correct ZONES for the Stop from a zone file of predefined geographical zones. If the zone file does not exist then the Stop will be rejected. See "Zonefile" on page 3 - 30.

3.2.3.1.12 Jobs and Job Order

This Stop can be grouped with other Stops to form a Job, all of which will be assigned to the same Resource. The JOB field is used to specify the Job to which the Stop belongs, and the JOB_STOP_ORDER field is used to specify where this Stop fits into the execution sequence of the Stops in the Job.

3.2.3.1.13 Job Priority

If the Job Stops are to be completed as quickly as possible without any other Stops being interspersed in the sequence, then this Stop (or any other Stop in the Job) should have its SINGLE_LINK flag set.

3.2.3.1.14 Service Level Agreement Priority

The field SLA_PRIORITY specifies the flat cost that will be applied when Jobs are done outside the Access Time Window.

3.2.3.2 Stop Service Level Agreement and Access Time Window

Planner 9.0 classifies Stop TWs as AccessTWs and SLATWs. AccessTW is the time within which the Job can be done at any time, whereas SLATW is the preferred time for doing the Job. SLATW and AccessTW are independent window-systems and a valid Stop must at least have one AccessTW. Oracle Real-Time Scheduler allows Stops to start as early as the AccessTW early time, while recognising that the Stops should preferably be started (or completed) within SLATW. The Stop Time Window Data Packet carries mandatory information that uniquely identifies the Stop Time Window. It defines a Resource access period. There must be a separate Stop Time Window Data Packets for every access definition — Stop Time Windows cannot be used multiple times.

Example 1: A store is open Monday to Friday stop from 09:00 to 17:00, delivery is preferred Monday morning from 10:00 till 12:00 or Friday afternoon from 15:00 to 17:00.

Oracle Real-Time Scheduler will represent this scenario with five AccessTWs for a week, each from 09:00 to 17:00, and two SLATWs: Monday from 10:00 till 12:00 and Friday from 15:00 till 17:00.

Example 2: A customer is at home all day, but prefer delivery not to be done around lunch time.

Planner will represent this scenario with AccessTW 09:00 to 18:00 with two SLATWs: 09:00-12:00 and 15:00-18:00.

3.2.3.2.1 Preference

The arrival of a Resource outside the specified AccessTW interval can be discouraged more or less by changing the LATE_COST data field value. Where several Stop Time Windows are associated with the same Stop, the preference for each window can be set using the WINDOW_COST data field.

If these fields are not set, no special preference will be given to the Stop Time Window.

3.2.4 For Scheduler Control

The Cost Item Data Packet sets the various weights (costs) used by Planner Scheduler. Planner Scheduler uses these costs to assess the usefulness of any changes it makes to the arrangement of Shifts and Stops.

The cost values included in the Cost Item Data Packet are called “Global” costs because they are applied equally and universally to the solution regardless of the object (Resource, Shift, Stop etc.).

Note: Most of the objects (Resource, Shift, Stop etc.) can specify “Relative Cost Multipliers” that are applied to the “Global” costs to alter the severity of the applied cost under particular circumstances. For example, a motorbike is faster and cheaper to run than a 10 tonne truck.

When Planner Scheduler commences, all Global cost parameters will be initialized to sensible values. If a Cost Item packet is received by Planner Scheduler, the values contained in the packet will override the current values of the particular global cost parameters. An empty field leaves the Global cost parameter value unchanged.

For a full explanation of the use of the cost parameters to “Tune” the performance of Planner Scheduler so that it reflects the goals of the operation as accurately as possible, refer to the companion manual entitled “Planner Users Manual”.

3.2.4.1 Cost Control

This table lists all the Global cost variables controlling the behavior of the various cost functions. Exponential costs derive a curve such that "ramp=A at x=0, doubling every B units".

Field	Cost Type	Description
Late off	Exponential	Costs units/sec after logoff.
Stop late	Exponential	Costs units/sec after stop access time-window.
Depot late	Exponential	Costs units/sec after depot access time-window.
Stop service window	Flat	Costs units/sec outside of stop service time-window.
Depot service window	Flat	Costs units/sec outside of depot service time-window.
Distance	Flat	Costs units/m of travel.
Travel Time	Flat	Costs units/sec of travel. Cost of each second the driver spends "working" or taking break. Time spent at depot, logon, logoff (unless on break) is excluded. Any delays at Depot (loading, waiting) is not costed. Time spent on Breaks at Depot are costed TravelTime is costed on a "run" basis, i.e. Cost is applied to the time between departure from LogStops and Depots to the arrival at the next LogStop or Depot. When there is a break between the previous stop and the Depot/LogStop, any waiting-time for the break is not costed, as it can be spent at depot/logoff. The latter is to prevent costing of time spent "waiting" for a break at depot.
Idle Time	Flat	Costs units/sec of idle time.
Max Weight	Exponential	Cost against load/max ratio
Max Volume	Exponential	Cost against load/max ratio
Depot window capacity volume	Exponential	Cost for exceeding depot-time window maxCapacity
Depot window capacity weight	Exponential	Cost for exceeding depot-time window maxCapacity
Zero capacity	Flat	Cost for zero depot capacity
Max. capacity	Flat	Cost for maintaing maximum depot capacity

Field	Cost Type	Description
Depot below target	Flat	Cost for depot below target
Depot above target	Flat	Cost for depot above target
RsrcAttr	Flat	Cost of allocation-mismatch
JobAttr	Flat	Cost of single job attribute conflict
Stop zone	Flat	Cost of stop not being in shift-zone
Round zone	Flat	Cost of none of the stops in a round being in shift-zone.
Shift	Flat	Cost of using the shift
Vehicle	Flat	Cost of using the vehicle
Longhaul	Flat	Cost of using a Shift, which is part of a LONGHAUL and is not terminating at the normal log-off position. At LOGOFF position for STANDARD or ONE_WAY shifts.
Access Window	Flat	Cost for access time window
Run	Flat	Cost the number of depot-stops in a trip
Site	Flat	Cost against using unnecessary vehicles at the same site.
Pickup service	Exponential	Cost applied to PKUP, DIST stop types and STOPs where load is picked up. Positive values bias towards early Collection, negative towards late Collection.
drop service	Exponential	Cost applied to DROP, COLL stop types and STOPs where load is delivered Positive values bias towards early Distribution, negative towards late Distribution.
Run separation	Exponential	Cost against relVolume. Applies a cost to loading goods during a Distribution-run.cost = fABC(l/c) * r;l = load added at stop.c = capacity available on resourcer = remaining stops within the run.
Lifespan	Exponential	Cost for late delivery. Cost as a function of the number of seconds a product is on board longer than its life-span. This cost is mutually exclusive with Max_Runlength cost.
Allocation	Flat	Cost against not allocating a stop
Haul	Flat	Cost of using a haul.
Site separation	Flat	Cost site separation
Product depot	Flat	cost of using product at the particular depot
Apex	Flat	Cost of not finishing a run at the Apex
Round length	Flat	Costs the length of the "round"
Round length variance	Flat	Costs the variance of the "roundLength"
Round	Flat	Costs the size of the round-area.
Round variance	Flat	Variance of the round.

Field	**Cost Type**	**Description**
Round overlap	Flat	Costs overlap in round.
Round end overlap	Flat	Costs overlap in the areas around the last stop of the round.
Load level slop	Flat	Costs deviations from average value/shift
Load level max	Flat	LoadLevel cost at value=0
Waiting	Flat	Costs the idle time of the driver
Past Waiting	Flat	Costs the time that the driver has been idle
Empty travel	Exponential	Costs associated with ratio between empty run and job's length
Break Early	Exponential	Costs associated with taking break early
Break Late	Exponential	Cost associated with taking break late
Shift Reserve Capacity	Exponential	Cost associated with exceeding shift reserve capacity

3.3. Operational Scenarios

The previous section describes the mandatory data fields for all the Planner objects, and shows how the Planner solution would be influenced by the use of the optional data fields available in the objects.

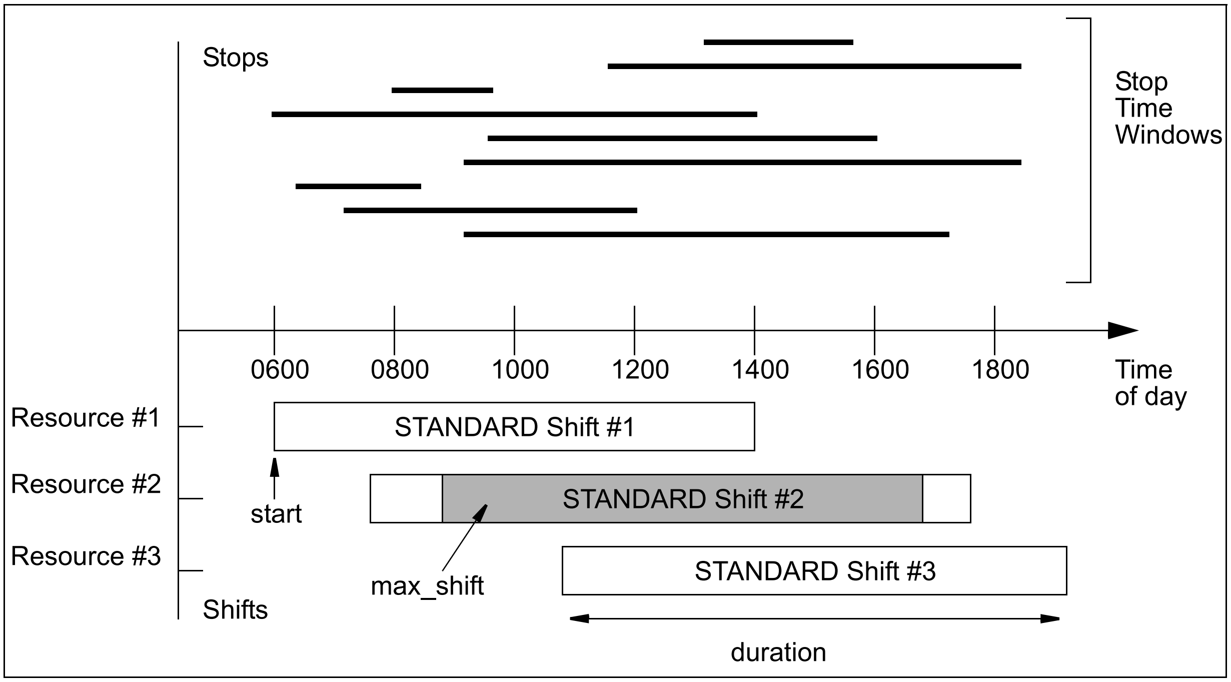
Now we can investigate how to set up the Planner data to solve some real scheduling problems.

3.3.1 Single / Multiple Day Scheduling

Setting up the required Shifts is one of the most important task, and sometimes the most difficult task to be accomplished. The working period of a single Shift is determined basically by the cost involved, by driver availability or by other scheduling requirements. For successful scheduling, the Shifts created must match the availability of Resources with the Collection and Distribution requirements of the Stops.

3.3.1.1 What type of Shift?

For a scheduling task that has all its work completed in a single work day, the only Shift Types needed will be the STANDARD or ONE_WAY Shifts, one for each Resource in the fleet. By using a combination of START, DURATION and MAX_SHIFT, each Shift can be set up to accommodate the requirements of simple operations.



Notice in the above illustration that Shift #2 allows Planner Scheduler the flexibility to schedule the Log-on and Log-off Stops at the times that are best suited to the solution.

Each Resource works its allotted hours and returns empty to be garaged at the end of the Shift.

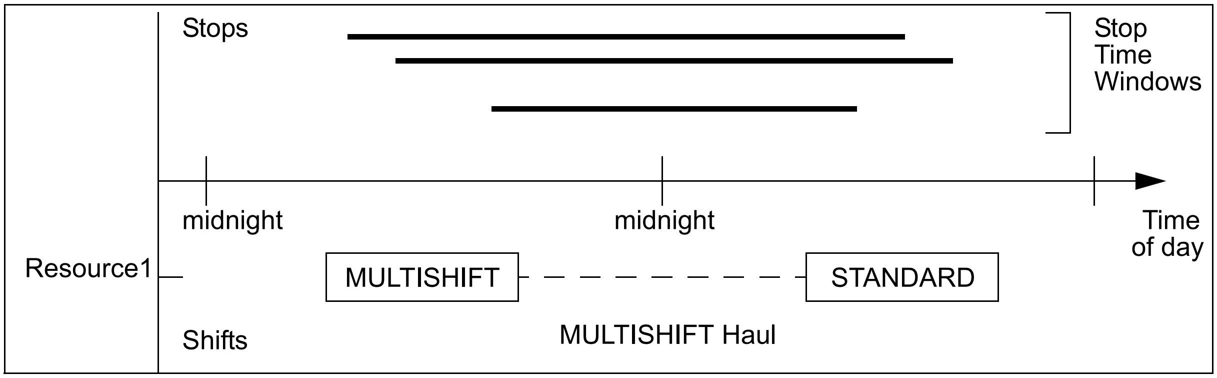
For a scheduling task where continuous operation is required, or where goods might remain on board a Resource beyond the normal work period, MANYSHIFTs, MULTISHIFTs or LONGHAULs will be needed. But how many? This question can only be answered by examining the operational data for the Stops and making an estimate of the required Load Horizon and Scheduling Horizon.

3.3.1.2 Load Horizon

The Load Horizon is the most distant delivery time for a load. It determines the minimum number of Shifts that need to be linked together for transportation purposes.

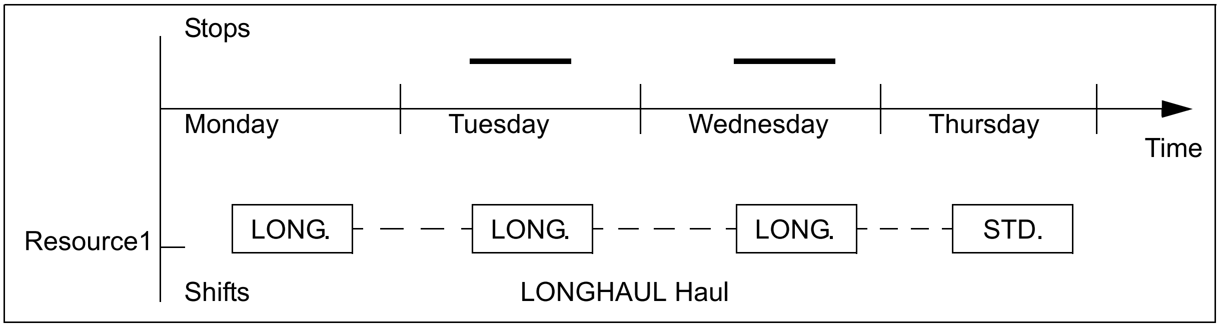
For a single work day operation the load horizon is about eight hours, so single shifts with standard working hours are satisfactory. If necessary the starting time of the Shifts can be staggered to encompass Collection and Distribution requirements that may lie just a little bit beyond the limits of standard working hours. (See previous illustration)

In the case of a local or regional operation where a load picked up today cannot possibly be delivered before tomorrow, then the Load Horizon may be 24 to 48 hours during which the load must remain on the Resource. This means that several Shifts must be combined into a MULTISHIFT Haul. The Resource works during it’s allotted hours each day and returns to be garaged at the end of the Shift. At the end of its MULTISHIFT Shift, it will have goods remaining on board.



For a long-distance transport operation where the transportation task itself takes a long time to be completed, the location of the Resource at the end of a day’s work may be unknown — perhaps requiring an overnight delay at a place that is not the specified Log-off position. Here the load horizon is not necessarily determined by the Stop Time Windows but by the travel time required to reach the Stops.

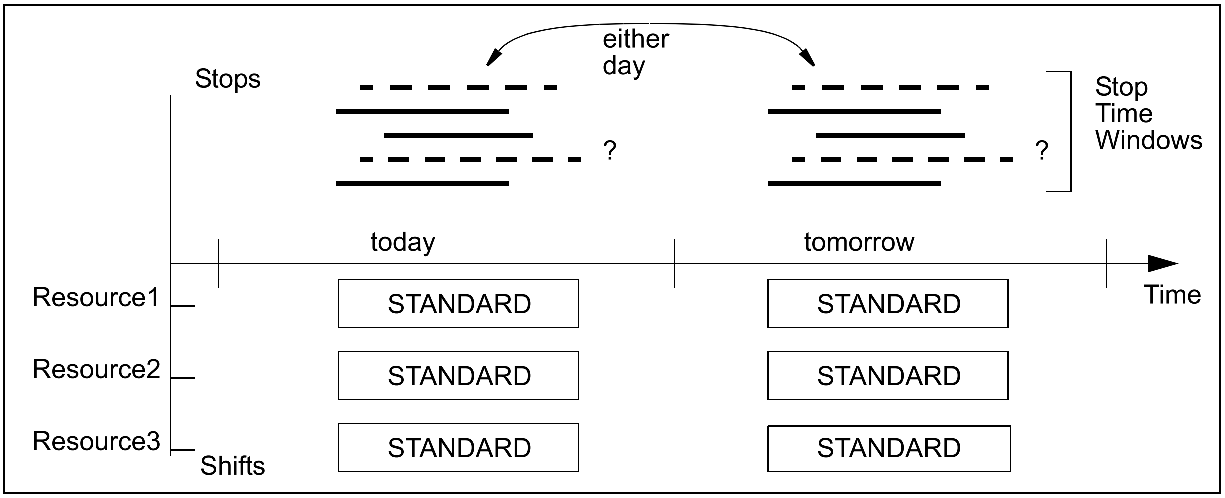
For Planner Scheduler to properly schedule these tasks, LONGHAUL Shifts will be required.



3.3.1.3 Scheduling Horizon

The scheduling horizon is the most distant time beyond which the knowledge of future work will not effect the current scheduling task. It determines which data and Shifts should be present during scheduling. It basically identifies how far into the future Planner Scheduler should look (today only, tomorrow, whole week) in order to make sensible scheduling decisions.

For example, if the work load for today and tomorrow is well known, and some work can be done either today or tomorrow, then it will be beneficial to allow Planner Scheduler to decide which work will be done today and which will be done tomorrow. This means that the Shifts for both today and tomorrow must be known to Planner Scheduler, even though only today's work will actually be undertaken by the Resources.



3.3.1.3.1 But Not Too Far Ahead

So, if the data to be scheduled is well behaved and well known for weeks in advance, then if the Shifts are set up, Planner Scheduler can schedule all the data in one go. Right.?

Wrong! Well, almost.

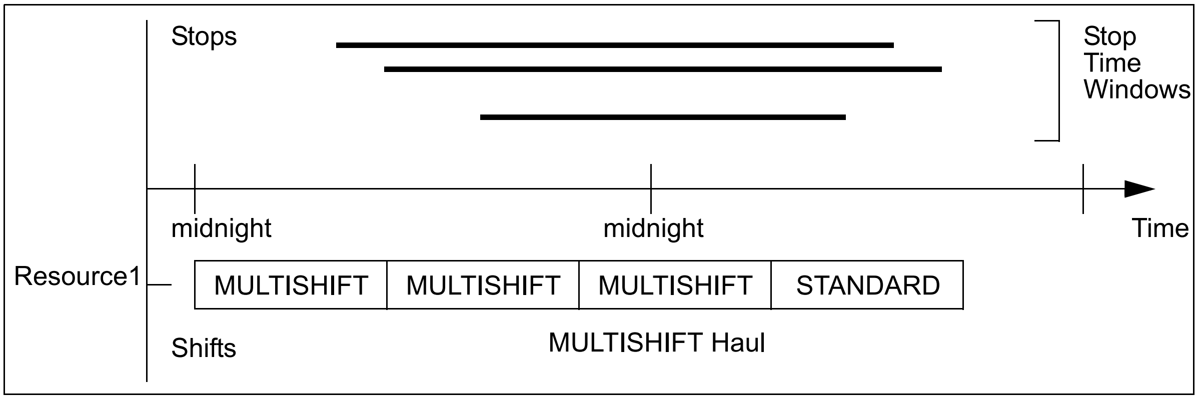
Planner Scheduler would have a good try at scheduling all the data, but it would be a terrible waste of computer time. Future events are unpredictable — drivers can get sick, trucks can break down, customers can change their mind. Then there is the hassle of printing and separating the manifests for each day's work. The larger the data set, the longer Planner Scheduler will take to reach a solution. If too much future data is included in the data set, then Planner Scheduler may not be finished by the time today's solution is required!

Oracle Corporation strongly recommends that the load and scheduling horizons be used sensibly. Only schedule data within these horizons, even though the requirements for future scheduling are known.

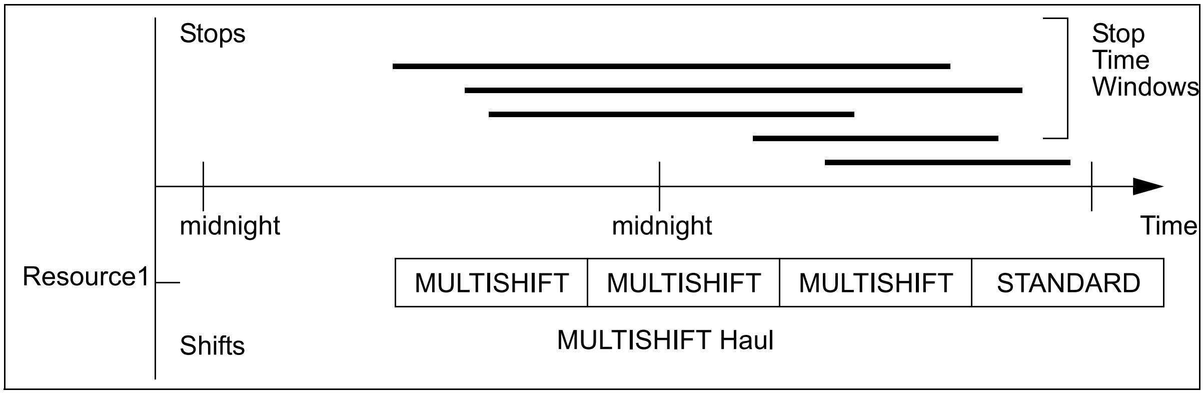
3.3.2 Continuous Operation

Where a transport operation runs around the clock, then both the Load Horizon and the Scheduling Horizon will extend into the future, and the Shifts must be set up to continuously span the Load horizon and to allow for driver change-over locations.

The Shift types that are used will be determined by any operational requirement that goods may or may not remain on board the Resource at the end of each Shift. It will be a STANDARD or MANYSHIFT Shift if no goods remain on board at the end of the Shift, otherwise MULTISHIFT or LONGHAUL Shifts can be combined into a HAUL.



Whenever the first Shift has been completed it can be deleted, and a new Shift can be added to extend the Scheduling Horizon as required.



To provide for driver change-over points, the Log-on location for a Shift should be the same as the Log-off location for the previous Shift.

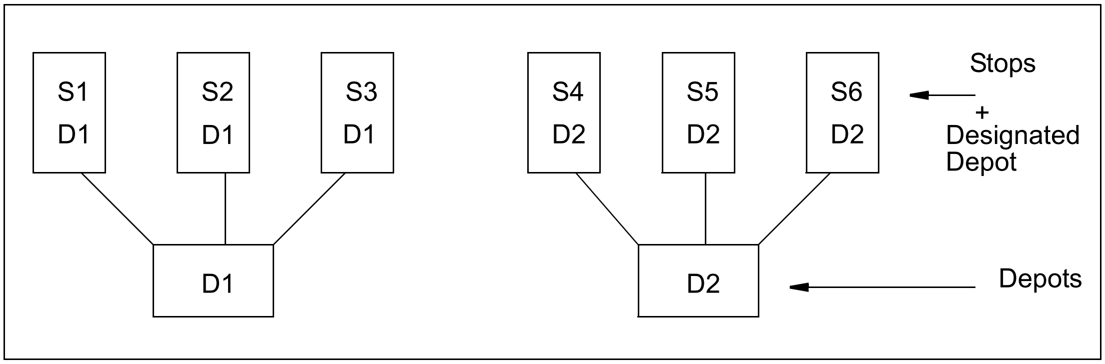
Refer MAX_HAUL on page 7 for further details on continuous operations.

3.3.3 Depots

For a Collection or Distribution problem where there is only one Depot to service all the Stops, the solution is usually straightforward and is solved by minimizing travel time, distance, overtime, or all three — depending on the goals of the enterprise. A physical Depot is regarded by Planner Scheduler as having a single function — either Collection or Distribution.

The Depot Access Time Window(s) and or Depot SLA Time Windows associated with the Depot controls the access of the Resources to the Depot. Planner will schedule the Resources to arrive at the Depot within the specified Depot Time Window(s). The Depot AccessTW and SLATW are similar in function to the Stop Service Level Agreement and Access Time Window discussed under section 3.2.3.2 on page 3 - 16.

If the loads that can be supplied by or delivered to a Depot are very specific, then each Stop can specify the particular Depot that must be used for Collection or Distribution tasks according to the nature of the loads.



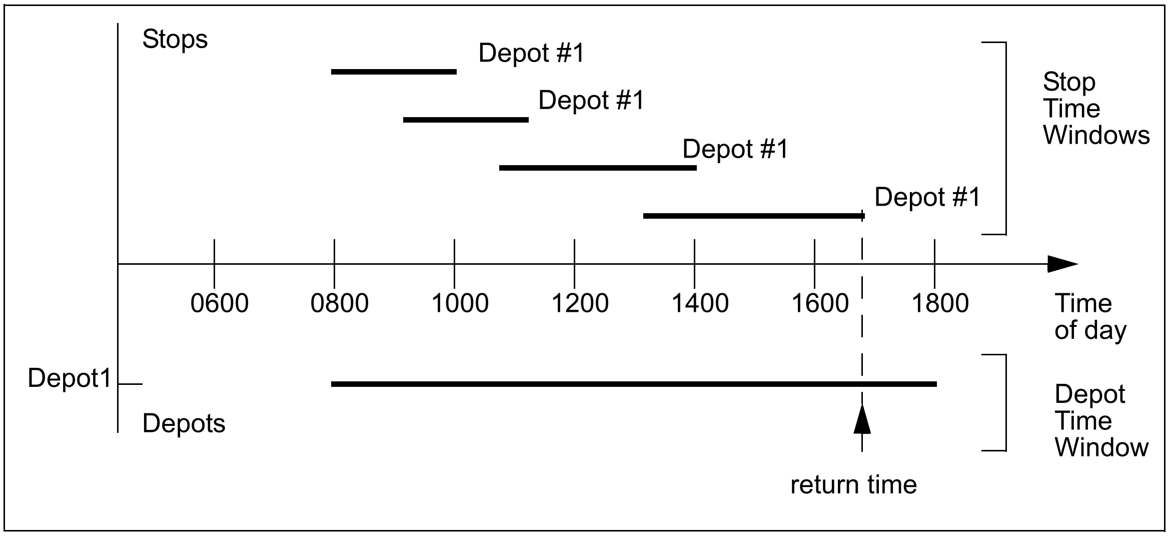
In situations where a Depot has multiple AccessTWs and or SLATWs, there is no way to tell Planner Scheduler which Depot Time Window should be used for a particular Stop — it will simply use the most convenient window. This problem is overcome by using “virtual” Depots.

Note: Wherever SLATWs are also specified for Depots, Oracle Real-Time Scheduler will attempt to use SLATWs over AccessTWs.

3.3.3.1 Virtual Depots

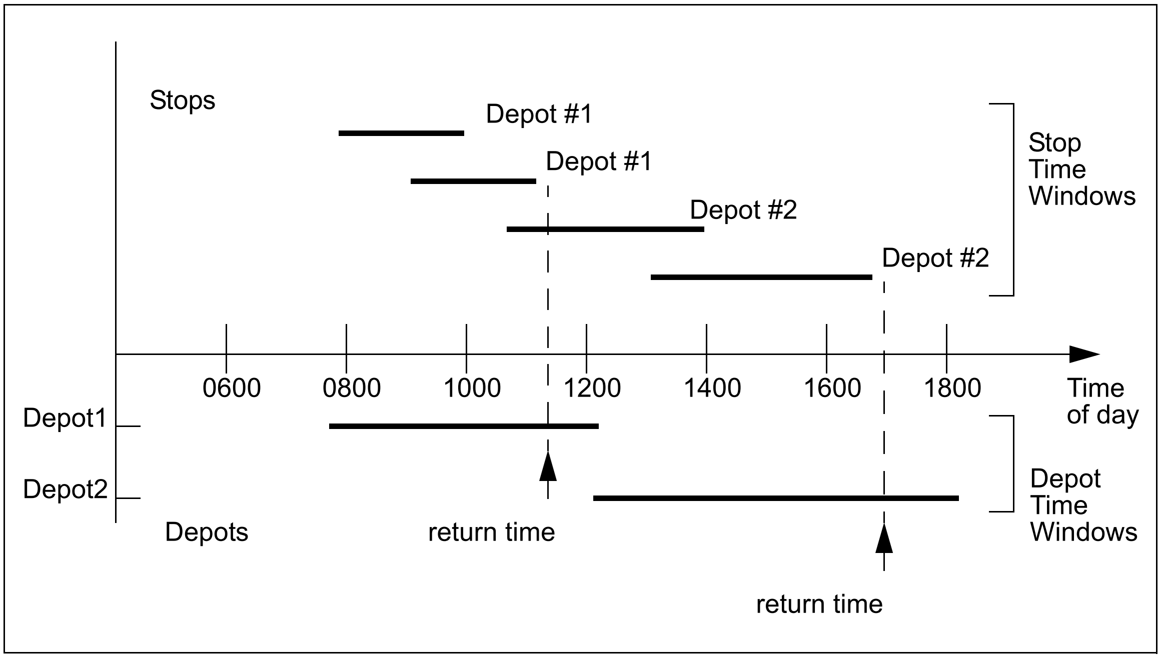
A Depot can be split into any number of “virtual” Depots in order to solve some scheduling problems. The virtual Depots, of course, all have the same physical location, but each virtual Depot can have its own set of independent Depot Time Windows.

This means that if a Stop is specified to use a virtual Depot, then Planner Scheduler will schedule the Stop to arrive within the Depot Time Window of that virtual Depot. This arrangement can be used to make sure that loads are returned to or distributed from Depots in an orderly manner. Consider the following Collection task, where all the Stops must be handled by Depot #1:



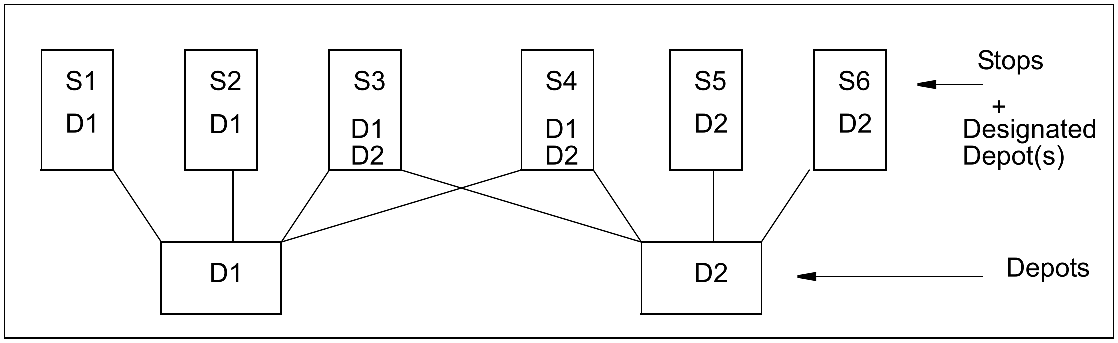
The most likely solution is that a Resource would pick up all four Stops and return to the Depot at about 1700 hours.

But what if it was necessary to have some goods back at the Depot by 12:00 hours.? Using multiple Depot time Windows would not work because Planner Scheduler cannot link Stops to specific Depot Time Windows. The solution is to split the Depot into two virtual Depots, and set the Depot for the Stops as follows:



3.3.4 Multiple Depots

Where there are a number of similar Depots, either real or virtual, that can be used to service the Stops, then it is important to allow Planner Scheduler the flexibility to choose the most appropriate Depot for each of the Stops. This is done by specifying in the Stop all the Depots that can be used by the Stop.

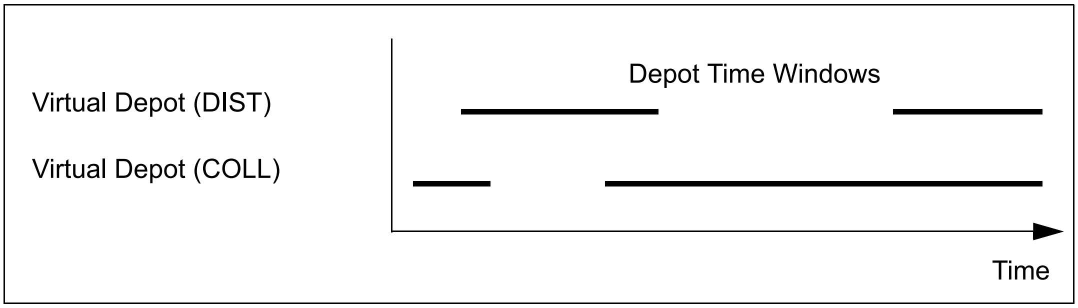


3.3.4.1 Combined Distribution and Collection

A physical Depot is regarded by Planner as having a single function — either Collection or Distribution. In practice, however, the physical Depots can often perform both functions simultaneously.

For Planner to handle multifunction physical Depots, there must be separate Depot Data (that is, a “virtual Depot”) for each operation at the physical Depot. The Depot location, of course, will be exactly the same for both virtual Depots.

Each of the virtual Depots can have entirely different Depot Time Windows specifying access by Resources. The separate windows can be used to stagger the arrival time of the Resources for each task if there is a bottleneck at the physical Depot entrance.



3.3.4.1.1 Run Separation

Run Separation is advantageous when a resource must be restricted to either Collection or Distribution runs. For instance if the resource in question is a rear loading one then care must be taken against Collecting goods which may obstruct future Distributions.

It is usually acceptable that a Resource can complete both a Collection Run and a Distribution Run during a single Shift. To ensure that the Stops of the separate tasks do not get interwoven, the RUN_SEPARATION Global cost parameters should be adjusted to provide Planner with the required amount of encouragement to separate the two tasks in the solution.

3.3.5 LTL Operations

For “Less Than Loaded” (LTL) operations, as the name implies, the capacities of the Resources are not regarded as a particularly critical boundary for the scheduling problem, although they must be taken into account while scheduling. For example, a motorbike could carry a long pipe, and a tray truck could carry an envelope — but of course they should be scheduled the opposite way.

A typical example of LTL operations is a Courier company that will have a large number of available Resources that can be used as the need arises. The goal of the Planner is to make sure that all of today’s work is completed on time. There is usually no penalty to the operation for using additional Resources and Shifts, so the Planner Global costs for RSRC, SHIFT and RUN would be very low.

The most important aspect of LTL is usually customer service. The distance travelled by the Resources is not as important as completing the Stops at the required time, so the

Planner Global costs for TRAVELDIST and TRAVELTIME would be low, while the STOPLATE Global cost would be very high.

Because of the wide variety of goods and Resources used by this type of operation, the proper specification of Resource and Job attributes is critical to success. There is no point in sending a small car to deliver a pallet of goods, even though it may fit if the job specified a Tail-lift to handle the weight. To encourage proper attribute matching, set the Planner Global costs for JOB_ATTR and RSRC_ATTR high, and make sure that the proper attributes are set.

Drivers are often contractors who get paid according to the work they do rather than the hours worked, so load-sharing amongst the fleet may be a more important consideration than reducing overtime. In this case, the Planner Global cost for OVERTIME would be low.

To make the best use of the available fleet, it is important to properly describe the abilities and preferences of each individual Resource. To do this, use the RELSPEED, ZONE_MUST and ZONE_PREF variables in the Shifts to highlight the relative differences between the Resources.

3.3.6 Full Load Distribution

Where a transport operation involves the delivery of the entire load capacity of a Resource, then the goals of the scheduling may be the efficient use of a fixed fleet of Resources and the proper usage of Resources that have different carrying capacities.

In this case, the Resource capacities may be critical so it is important to properly specify MAXWEIGHT, MAXVOLUME and/or MAXSIZE for the Shifts, as well as WEIGHT, VOLUME and SIZE for all the Stops.

The actual costs associated with operating large vehicles may be quite high, so there may be a real saving to the operation if fewer Resources can be used. Here, the Planner Global costs for RSRC and SHIFT would be high. For a company fleet, it may also be necessary to discourage too much overtime, so the Planner Global cost for OVERTIME would be adjusted until an acceptable level of overtime is achieved in the solution.

In the case where third party contractors are used to handle peaks in demand, then the RSRC_COST and SHIFT_COST multipliers for the contractor Shifts would be set high so that the company fleet would always be used first.

3.3.7 Milk Runs

The transportation task of some operations is so well behaved and well known that the same schedule solution can be used again and again. These schedule solutions are often referred to as “Milk Runs”, and are usually associated with very narrow time windows at the Stops.

The problem with Milk Runs is that they are used so often that Allocators and customers often think that they are set in concrete, and that there is no other way to solve the problem. The narrow time windows have usually evolved historically, and should be re-evaluated at regular intervals instead of shoe-horning new Stops in between existing time windows. In many cases, the customers are quite happy for the Resource to arrive at some other time, as long as they know what that time will be.

To re-evaluate a Milk Run, find out which Stops can have flexible Stop Time Windows and make these Stop Time Windows as wide as possible before scheduling the data set.

3.3.8 Strategic Planning

Strategic planning is usually required when an operation reaches a condition that is unacceptable — for example, too many late deliveries, too much overtime, depots can't handle the demand.

Planner can be used to help the Allocator make some business decisions, such as:

- Where should I locate a new Depot.?
- If I had one more Resource, would the problem disappear.?
- What would happen if I combined my two operations.?

and more.

The answers are found by examining the schedule solutions that are produced when the same data set is repeatedly scheduled under slightly different conditions.

For example, to find a good location for a Depot, run several schedules with the Depot in a different location each time, then examine the total travel time of each solution to see which position makes most efficient use of the fleet.

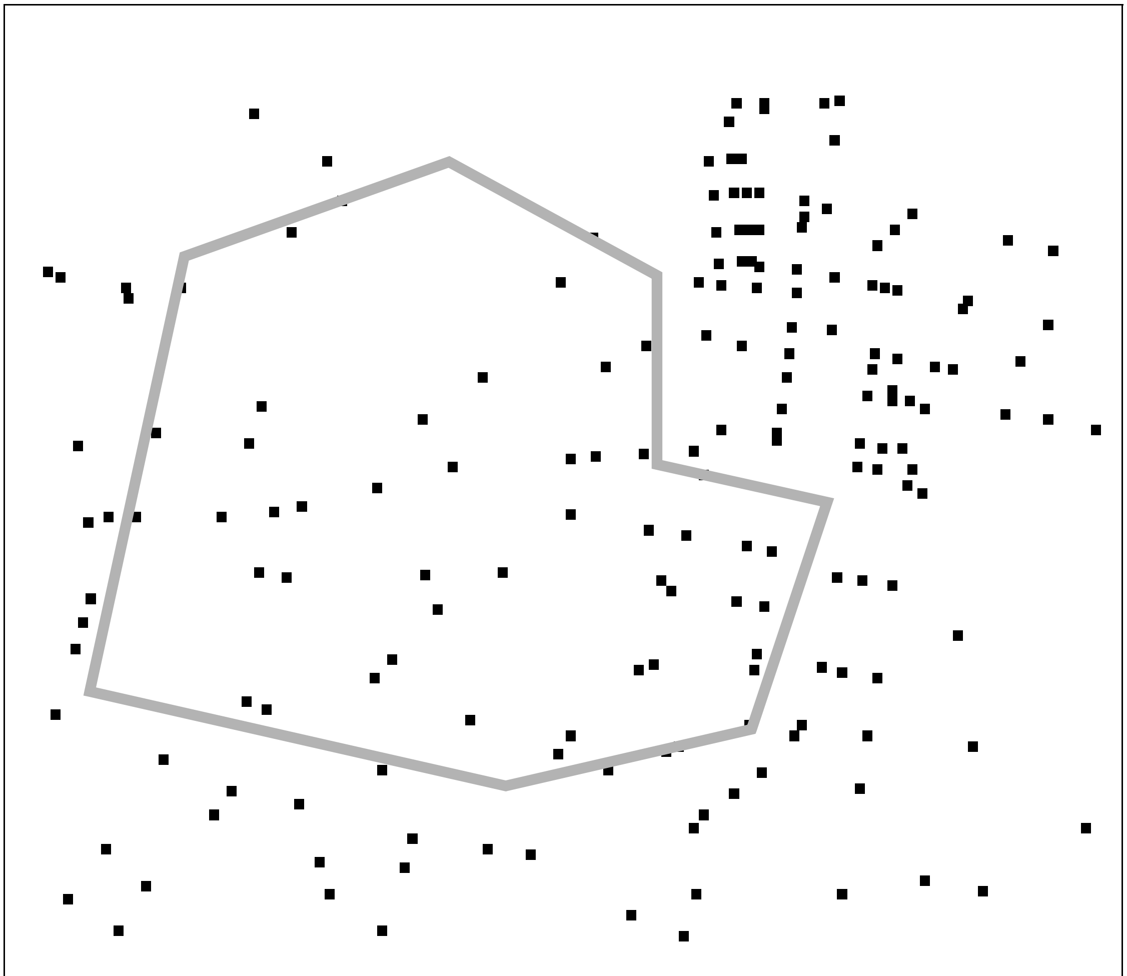
For example, if the fleet is too small, add Resources to the fleet and observe the changes in the number of late Stops or the amount of overtime. Alternatively, if some Resources are under-utilized, increase the Planner Global costs for RSRC, SHIFT and RUN to see if some Resources can be eliminated altogether.

For example, scheduling two independent operations simultaneously may show that better performance is possible if Resources can be shared between the two tasks.

Also see the companion manual "Planner - User Manual" for a description of "tuning" the solution.

3.4. ZONEFILE

If the task to be scheduled has any geographically zoned operations, many scheduling problems can be solved by making these zones known to Planner. A maximum of 256 zones can be defined. A zone is defined as an imaginary polygon that is constructed of coordinates (usually Longitude/Latitude) within the map space known to Planner.



Polygon created with a set of longitude/latitude coordinates.

The polygons that make up the zones are described and named in a zone file in the Planner installation. This file is read from the disk each time Planner is started, and can be used to automatically assign a geographic zone to incoming Stops. It can be edited at any time to alter the composition of the zones by either using a text editor which may be cumbersome or more conveniently using the Planner Map Window tools. Refer the companion Planner User Manual for details on creating, editing and deleting zones.

Note: The default filename is *zonefile*, but can be altered at run time by the parameter setting in the config file.

3.4.0.1 Automatic Zoning

If pre-defined geographical zones exist, Planner can automatically associate incoming Stops with the Zone(s) in which they lie, so that Planner can subsequently recommend Resources with matching Zone preferences.

For automatic zoning to occur, the incoming Stop data must have its ZONE field set to AUTO. If automatic zoning is to be used for one or more Stop, then it is usually preferred to use it uniformly for all Stops in the solution.

3.4.0.2 Zone File Format

The format of the “zones” file is plain ASCII text. Blank lines are permitted. Any line that has ‘#’ as the first character will be ignored. A zone definition must be started with “zonename=*identifier”*, followed by keyword “zonestart”. A list of up to 64 coordinates can be specified. The definition must end with the keyword “zoneend”. A typical format is as follows -

```
zonename=identifier
zonestart
    colour
    visible
    x-coord 1,y-coord 1
    x-coord 2,y-coord 2
    .
    .
    x-coord n,y-coord n
zoneend
```

where:

identifier-	Identifier of the zone (32 chars max)
x-coord N-	x axis value of the coordinate
y-coord N -	y axis value of the coordinate
colour-	alphanumeric value identifying the colour and it is set from Client
visible-	TRUE/FALSE values can be toggled in Client

Note: All text printed in *italics* should be replaced with user defined values. No space exists between the coordinates, just a comma character.

For example, the file might contain:

```
# this is zone definition

zonename=CBD
zonestart
colour=ff55ffff
visible=TRUE
    145.031628,-38.273317
    145.137305,-37.607252
    145.011605,-37.829345
```

```
        145.055078,-37.829657
        145.313851,-38.213849
        145.159066,-37.904938
        145.017774,-37.902750
        146.962833,-38.565041
    zoneend
```

The first line will be ignored by Planner as it has ‘#’ as the first character. The zone created from this file will be identified as “CBD” and the eight coordinates specified will form an imaginary polygon (zone) on the map. In this example, it is assumed that the map space is calibrated in Longitude and Latitude. The maximum number of coordinates used to create a zone is limited to 64, any additional coordinates specified will be ignored. If, for any reason, a zone definition is not understood by Planner, the zone will be discarded.

Dividing geographical areas into zones is completely optional and is only necessary when implementing zoned operation.

Chapter 4: Interfacing to Planner

This chapter describes the Data Packets and the protocol required when communicating with Planner via user-written Gateway software.

4.1. DATA PACKETS

Data communication between Planner and the Host system always takes place using Data Packets.

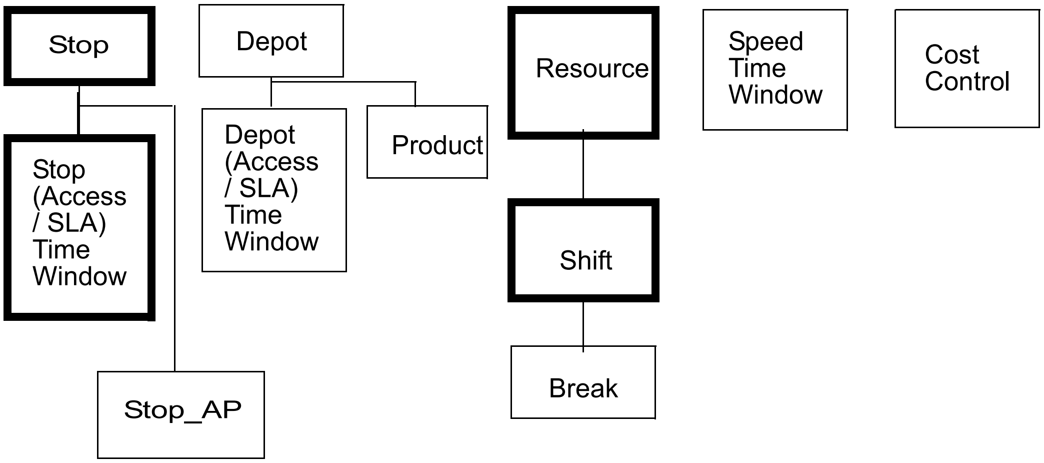
4.1.1 Basic Data Packets

The basic information that is supplied to Planner for creating schedule solutions is contained in the Data Object packets (Resources, Shifts, Stops, Depots, etc.). The data in the Data Object packets is "**Static data**" - it does not change as the schedule progresses.

4.1.1.1 Object Hierarchy

The Data Object packets that describe the Planner static data are not always independent and self contained. Some packets will need to refer to other data that is assumed to be already known to Planner.

The Object dependencies are as follows:



To undertake any scheduling task, Planner must receive, at the very least, the static data for the Resources, their Shifts and the Stops to be scheduled.

4.1.2 Other Packets

The data describing the current or Dynamic status of a Data Object is sent to Planner by the Data Update packets (and the Real Time packets, if licenced).

The interface can execute some Manual Scheduling operations using the Manual Scheduling packets.

The information that describes the solution to the problem is returned to the interface in Solution Object packets. These contain the work assignments and event times that actually make up the solution.

There are other packets that are used in the administration of Planner. These are called Mode Control, Status Reporting, Mobile Data packets, and Data Transfer packets. They are used to assist in the communication of data to Planner and to administer the operation of Planner.

4.1.3 Packet Construction

Data Packets are constructed using printable ASCII characters. The Data Packets are not all the same length, but are flexibly constructed to accommodate the data required for the particular purpose of each packet type. When the Data Packets are packaged up for communication, they must conform to the Host Interface Protocol (HIP) expected by Planner otherwise communication will fail. For a full description of the format of the Data Packets used by Planner, see Appendix A.

4.1.4 Packet types

The available Data Packet types are shown in the tables below. The Gateway interface should be able to generate or properly receive all the packets described.

Type	Packet Name	Direction	Use
Identification (static data)	CLIENT_TYPE	To Planner (via GUI Server if available)	Identifies the interface as a Client of Planner, and the Version of HIP Packet to be used.
Client Connection	CLIENT_CONNECT	To Planner (via GUI Server if available).	Request connection to Oracle Real-Time Scheduler directly or via GUI Server if set up so.
	CLIENT_REJECT	From GUI Server to Client.	Send from GUI Server to the second Client if connection request is made by two Clients with same ID.
Broadcast	CONNECT DISCONNECT	From Oracle Real-Time Scheduler	Broadcast Connection/ Disconnection status to all Clients.
User Access Control (static data)	USER_AUTH_REQ USER_CONNECT AUTH_RESPONSE AUTH_ALLOW AUTH_DENY	From Oracle Real-Time Scheduler	The User Access Control packets interact between the Client and Scheduler to determine the User's Access Level.
Report	REPORT_ON REPORT_OFF AUTH_COMPLETE	From Oracle Real-Time Scheduler	

Type	Packet Name	Direction	Use
Data View	VIEW	Bi-directional	Sets a particular data “View” for the interface. Also Reports the current view to the interface.
	VIEW_ALL	From Planner	Tells the interface all the data “views” that are available.
Data Objects (static data)	DEPOT DEPOT_TW DEPOT_SLA_TW RSRC SHIFT BREAK STOP STOP_TW STOP_SLA_TW STOP_AP PRODUCT_DEPOT SLOT SPEED_TW COST_ITEM STATUS_ITEM	Bi-directional	Static data. Basic data sent to Planner based on which a schedule will be created. Used by Planner to send data to the interface.
Solution Objects	PARAM_RSRC PARAM_SHIFT PARAM_STOP PARAM_DEPOT PARAM_DEPOT_TW	From Planner	Used by Planner to send data changes to the interface.
Mode Control	RESET	To Planner	Resets Planner to it’s start-up state.
	PLAN_INIT		Removes the current plan (if any) and generates an initial plan
	SCHED_ON SCHED_OFF		Activates or deactivates the Planner Scheduler.
	BATCH_ON BATCH_OFF	Bi-directional	Sentinel packets.
	COST_BATCH_ON COST_BATCH_OFF		
	SR_BATCH_ON SR_BATCH_OFF		
	CA_BATCH_ON CA_BATCH_OFF		
	BATCH_DONE	From Planner	Notifies the sender of receipt of all data and of completion of processing on the data.
	SR_BATCH_DONE		
	CA_BATCH_DONE		

Type	Packet Name	Direction	Use
	ACCEPT_CLIENTS	To Planner	Enables Scheduler to accept Client connection, when Scheduler is not started as a server (that is "serverMode=FALSE"). Scheduler receives this packet from Switch.
Data Update Objects	PLAN_RSRC PLAN_SHIFT PLAN_STOP PLAN_DEPOT PLAN_DEPOT_TW	To Planner	Used to update Planner objects.
Data Transfer	SEND_PLAN	To Planner	Request by the interface for a copy of all the current Planner data. Planner will send various Data Object packets and Parameter packets to the interface.
	SAVE_DATA	From Planner	Request by Planner to the interface to select from different data specifications for saving data onto the Host system. Or to retrieve data from Planner and save it to the Host System
	LOAD_DATA		Requests made by Planner to the interface (GUI) to select a data subset from a list of data specification. Or Request by Planner for data to be sent by the interface.
	SYNC_PLAN	To Planner	Request for Planner to send all schedule changes since the last SYNC_PLAN request. In response, Planner sends any changes to the schedule solution using various parameter packets. Static data will be sent only if it has been changed.
Mobile Data	REQ_TEXT_TO_ MDT	To Planner	Requests a message to be sent to the MDT
	TEXT_TO_MDT	From Planner	Sends a message to an MDT.
	JOB_TO_MDT		Sends a job-details message to MDT.
	BREAK_TO_MDT		Sends a response to Client Initiated Break Despatch Message.
	TEXT_FROM_MDT		Send text messages to Scheduler from MDT.

Type	Packet Name	Direction	Use
MDT RealTime Autodirect messages	GOTO_DEPOT	From Planner	Sends a depot-details message to an MDT.
	ARRIVED_DEPOT STARTED_DEPOT COMPLETED_DEPOT	To Planner	
MDT-RealTime messages	MDT_ARRIVED MDT_STARTED MDT_COMPLETED		Send Job completion details to Scheduler as the job progresses.
	MDT_ARRIVED_DIST MDT_STARTED_DIST MDT_COMPLETED_DIST		Sends the Vehicle updates at the Depot
	MDT_ENROUTE MDT_ACKMDT_CLRSTAT MDT_DEFERRED MDT_GO_HOME MDT_RESUME_WORK MDT_ACK_GO_HOME MDT_ACK_RESUME_WORK		Other status update requests sent to Planner to notify changes in statuses of Stops/Jobs.
	MDT_CURPOS MDT_LOGON MDT_BREAK MDT_DELAYED MDT_LOGOFF		Status updates from the Gateway to Planner with information on the position of the vehicle
Message Delivery Progress Status	MSG_STATUS	To Planner	Reports on the progress of a transmission.
Status Reporting	REPORT_ON REPORT_OFF	To Planner	Activates or deactivates continuous reporting by Planner to the interface. If active, Planner sends any changes to the schedule solution as they occur, using various Parameter packets. The static data is assumed to be known to the interface.
	STATUS_BATCH_ON STATUS_BATCH_OFF	From Planner	Sent spontaneously by Planner during automatic scheduling reporting current state of the plan and Scheduler.
	COMPLETED WARNING ERROR SYSTEM_WARNING		Sent spontaneously as a result of various event and conditions detected by Planner.

Type	Packet Name	Direction	Use
Manual Scheduling	STOP_DEL DEPOT_DEL SHIFT_DEL RSRC_DEL BREAK_DEL STOP_TW_DEL PROD_DEPOT_DEL DEPOT_TW_DEL STOP_SLA_TW_DEL DEPOT_SLA_TW_DEL SLOT_DEL	To Planner	Deletes Data Objects from Planner
	CHS_SHIFT CHS_SLOT SR_CHS_SLOT		Returns a list of possible single stop insertions in the schedule.
	CHS_REPLY	From Planner	The result sent from Planner in response to the above CHS-requests.
	CHS_REPLY_END		Planner will send no other reply packets.
	RSRC_POSITION RSRC_DELAY	To Planner	Requests Planner to note changes made to vehicle positioning and delays in the availability of vehicles
	SHIFT_DISABLE SHIFT_ENABLE SHIFT_CLOSE SHIFT_OPEN SHIFT_START SHIFT_COMPLETE SHIFT_STANDBY		Requests Planner to change the statuses of the shift/shifts accordingly.
	SHIFT_OPTIMIZE		Requests Planner to optimize a shift.
	BREAK_DISABLE BREAK_ENABLE BREAK_FREEZE BREAK_START BREAK_COMPLETE		Requests Planner to change the status of the Breaks accordingly
Manual Scheduling (continued)	RUN_CLOSE	To Planner	Requests Planner to close a Run.
	DEPOT_CUTOFF		Requests Planner to close all Runs starting before CutOff
	DEPOT_CUTOFF_CONFIRMED	From Planner	Planner confirms assignment of a Job

Type	Packet Name	Direction	Use
Manual Scheduling (continued)	JOB_DISABLEJOB_FREE JOB_DRIVER_ACK JOB_ASSIGN JOB_ALLOCATE JOB_DESPATCH JOB_DESPATCH_RUN JOB_CLOSE JOB_ASSIGN_COND CA_JOB_ASSIGN_COND JOB_ALLOC_COND JOB_TRANSFER	To Planner	Requests Planner to change the status of the Job (collection of Stops) accordingly.
	STOP_DESPATCH STOP_ENROUTE STOP_ARRIVE STOP_POSTPONE STOP_START STOP_COMPLETE		Requests Planner Scheduler to set the status of the Stops as requested by the Allocator or MDT/driver
	JOB_FIX_DEPOT JOB_FIX_HAUL JOB_FIX_RSRC		Request Planner to associate a particular Depot, Haul or Resource with a Job.
	JOB_LINK JOB_SEQUENCE JOB_UNLINKJOB_UNSEQ		Request to Planner^{TM} indicating restrictions placed on the order in which the stops are to be scheduled.
	JOB_INSERT STOP_INSERT INSERT		Indicates where to insert Stops/Jobs thus changing the order of Stops/ Jobs within the schedule.
	GROUP UNGROUP UNGROUP_JOBS		Groups/Ungroups Jobs thus making sure they are associated with the same shift/resource.
	BIND_STOP UNBIND_STOP SET_SUBWINDOW TIGHTEN_BOND		Request Oracle Real-Time Scheduler to perform Binding/ Unbinding operations on Jobs.
	GO_HOME RESUME_WORK		Send Go Home/Resume Work instruction to Oracle Real-Time Scheduler.
Real Time	REAL_TIME	To Planner	Enables real-time operations in Planner (if licensed)
	JOB_LOAD		Updates the load attributes of a Job.
Print Manifest Server requests	PRINT_RUN	To Print Manifest Server	Request from Client to Print Manifest for Run.
	PRINT_SHIFT		Request from Client to Print Manifest for Shift

4.2. WHAT IS A GATEWAY?

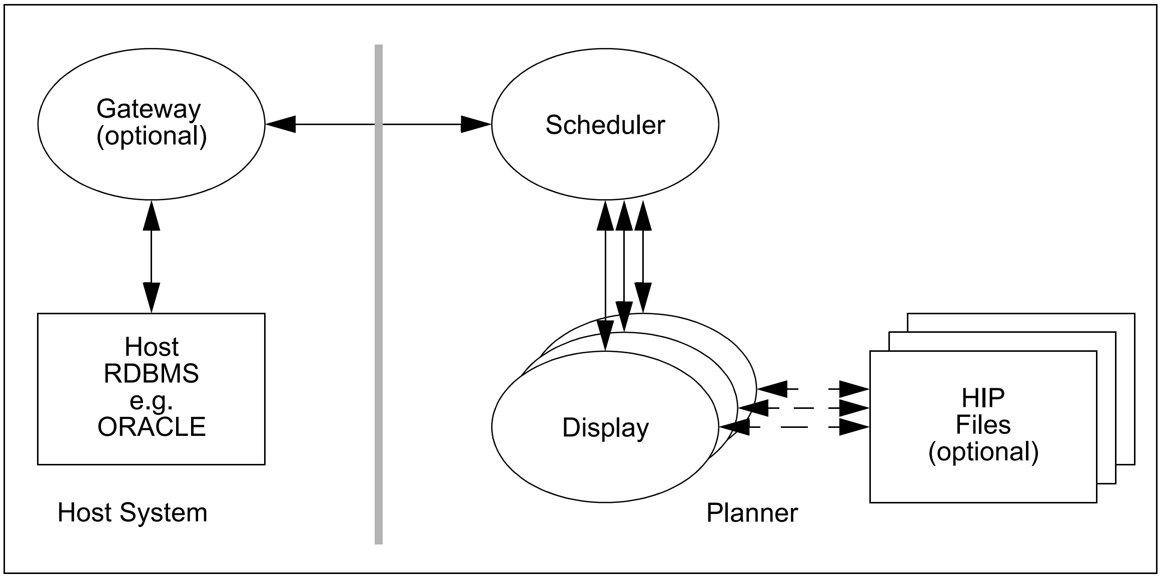
A Gateway is an interface program that must be written to provide a link between Planner and the computer (known as the Host system) that holds the information about the despatching task. The Gateway must be able to access, either directly or indirectly, the information in the database to retrieve data (spontaneously or on request from Planner) and send it to Planner asynchronously and in a “real time” manner. Gateway must also be able to return data to the database to ensure that the database information is current, and reflects any new information reported by Planner.

The Gateway is NOT supplied as part of the Planner software.

The purpose of the Gateway is to send operational data such as job and fleet information, from the Host database to Planner for consideration by Scheduler. The Gateway also receives messages back from Planner notifying job status changes for insertion back into the database, such as job allocation or completion messages.

4.2.1 Where Does Gateway Run?

The Gateway interface program is normally compiled and run on the Host computer. It communicates with Planner via an ethernet network link (TCP/IP) between the two computer systems.



Note: Other configurations are possible depending upon the capabilities of the available host computer and communication protocols. Installations can be tailored to suit many communication methods (by prior agreement with Oracle Corporation) but the following principles apply in all cases.

4.3. Gateway Objectives

It is the responsibility of the Gateway to keep track of the need to transmit data to Planner, to spontaneously transmit the required data and to handle any messages returned from Planner. The Gateway must be able to handle a continuous flow of data (in either direction) between the Host and Planner. This process should happen on a message-by-message basis, that is, the Gateway activity occurs in “real time”. It does not store sequences of messages for later processing.

The Gateway software should execute as a single process or daemon, and should be able to start and stop at any time without detriment to either Planner or the Host computer.

Also see Chapter 5: “Host Database Considerations” on page 5 - 13.

4.3.1 Sending Data to Planner

The Gateway is responsible for the timely transmission of the Job and Fleet information to Planner and handling messages that Planner sends in response. It should also be able to respond to any transmission errors or data rejections, allowing the Host System Manager to rectify the problem as quickly as possible.

4.3.2 Receiving Data from Planner

Planner will send data or messages to the Host as a result of a prior transmission from the Host or spontaneously as a result of a Scheduler action or as a result of a change in Job or Fleet status.

The Gateway should be able to rapidly handle the data sent by Planner so that long message delays do not occur, and thus ensure that all information stays current.

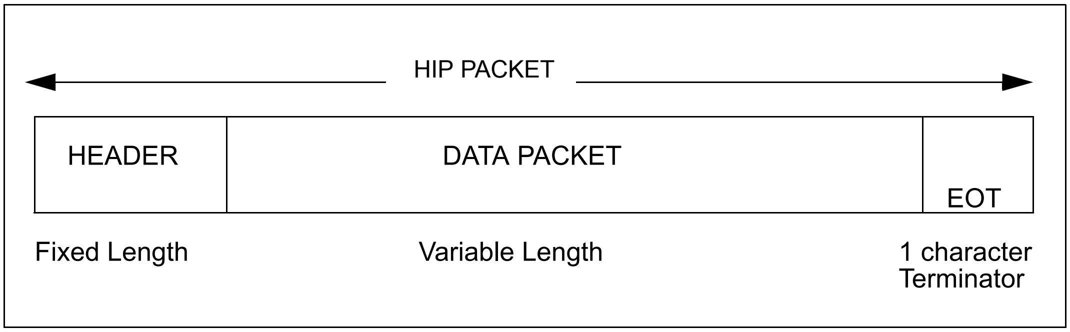
4.4. GATEWAY PROTOCOL

Communication between Gateway and Planner is accomplished using the Oracle Corporation proprietary Host Interface Protocol (HIP). The HIP for Gateway uses TCP/IP based ASCII messaging for data transfers. ASCII messaging was chosen as it is easily extensible, easy to use, verify and log. It is also machine architecture independent allowing a variety of customer machines to connect to Planner.

For successful communication, the HIP Packet messages must be built using only the Data packets described in Appendix A.

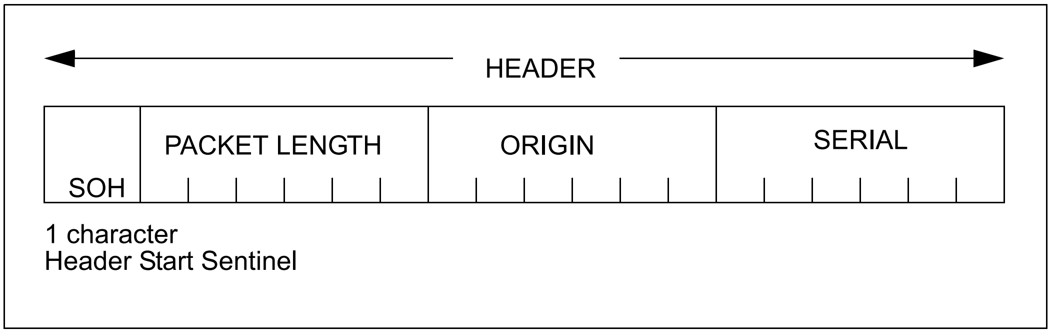
4.4.1 Building HIP Packets

Information is exchanged as packets of ASCII data (called HIP Packets), each packet consisting of a header, a Data Packet and a terminating character. The header for all packets has the same format and length, however the total packet length is variable. The format of the data component of the HIP Packet is related to the data being exchanged.



4.4.1.1 Header

Each HIP packet begins with a fixed length Header section that is used to identify this particular HIP packet and to specify the total byte length of the entire HIP packet.



where -

Field Name	Byte Len	Format	Description
SOH	1	Character	Start of header character (ASCII character 1)
PACKET LENGTH	6	Cardinal No.	Total HIP packet byte length including header
ORIGIN	6	Character	Identification of origin
SERIAL	6	Cardinal No.	Originators serial number for packet

4.4.1.1.1 Length

The PACKET_LENGTH field is set by Gateway to the total byte length of the entire HIP packet being sent. The length of the Data part of the HIP packet will be determined by the contents of the fields in it, so the PACKET_LENGTH value cannot be calculated until the Data part of the HIP packet is constructed.

4.4.1.1.2 Origin

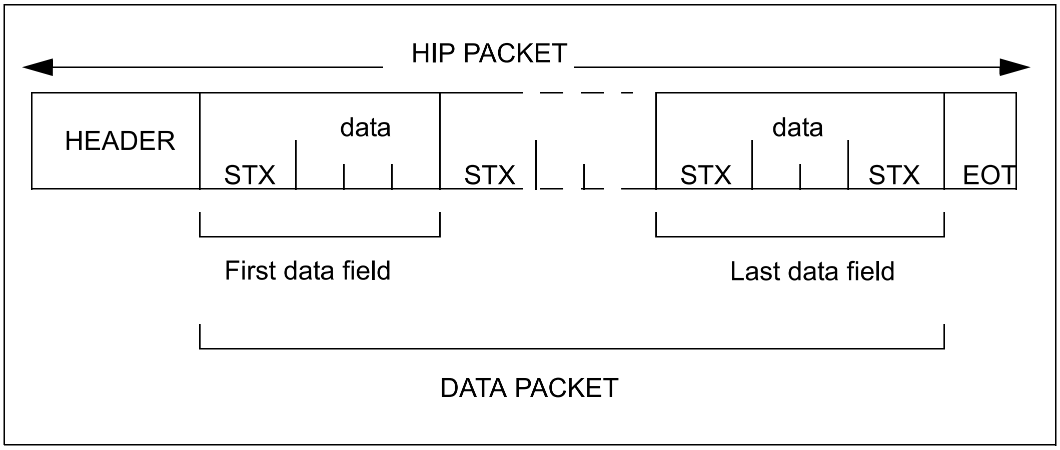
The ORIGIN Identifier is set by Gateway to an arbitrary character string that will be used for logging purposes only. To avoid any confusion, Oracle Corporation suggests “GATEWY”.

4.4.1.1.3 Serial

The SERIAL number should be incremented each time a HIP packet is sent by Gateway. This number is used for logging purposes only.

4.4.1.2 Data Packet

The data to be sent is placed between the packet header and the EOT character. Data fields within packets are of variable length, and to delimit fields within the packet each field begins with an STX character (ASCII character 2). The last field in the packet is also terminated with an STX character, that is, immediately preceding the EOT. An empty field must still have the appropriate delimiters.



Chapter 5: Real Time Interfacing

This chapter describes the Real time interfacing implemented in Planner allowing it to represent the states of a schedule just as it would occur in the real world.

5.1 REAL TIME ENVIRONMENT

When Planner is used in a Real time environment, it schedules data that is actually happening at that current time. Stops are scheduled for each vehicle/shift. The drivers are notified through their MDT's.

An ongoing communication between the driver and the Allocator or between the MDT and the gateway can be used to update the status of the job, vehicle and shift data objects.

Within a real-time environment, Planner must be able to represent the state of a schedule as it is executed in the real world, adjusting an existing schedule to accommodate new information.

Planner may accept or reject an update depending on data packet errors, inconsistencies and conflicts that may arise.

5.1.1 Real Time Customization

Gateway uses the Real Time packets to customize Planner's real time behaviour.

Real time features can be turned off, in which case Planner comes up with an optimal solution for all the Stops to be handled. The Allocator can then use the Plan to chart out itineraries for the drivers. Real time features may also be turned off when Planner is used for strategic planning, to check the effects of increase or decrease in the cost of various parameters - i.e when the Allocator uses the solution to study how the changes in various parameters can be used to reflect the business rules of the company.

With Real-time features turned on, the Gateway can interact with Planner to update the status of the various Stops, Shifts and Resources. These updates can be made by the Gateway automatically from the feedback obtained from the MDT's or based on the changes made by the Allocators on the Client interface.

Real time interaction can be setup in two modes, in the first mode, the status of the stops and the position of the vehicle are changed automatically as specified in the Plan. Changes are made automatically from the information obtained in the packets without actually waiting for updates from the vehicles. It simulates travel to next stop, notifies completion at the proper time, and then continues the simulated travel. This mode of real time interaction is called the **Auto Complete Mod**e.

But sometimes the schedule does not go exactly in accordance with the Plan, there may be reasons such as unexpected delays at a Stop, or weather conditions that may cause further delays, or change in the timings at which the stops are actually visited. The other mode takes these factors into consideration. It automatically advances the resource to the next uncompleted stop. It does not automatically update the status of the stops. It waits for an MDT update or a manual update by the Allocator on the Client interface before it marks a stop as COMPLETED. This mode of real time interaction is called the **Auto Positioning Mode**.

Planner has extended functionality called **Auto Direct** that automatically despatch the jobs to drivers. It enables Scheduler to select the next stop for a driver/MDT and to send the details about the next job to the MDT. This feature will only work in Plannerinstallation sites where MDTs are used for communication with the drivers. This feature can be activated using the command line parameters or from the launchPad (Please see the Planner Installation Manual for more details).

When Auto Direct is turned on, the sending of manual despatch messages will be disabled. The Allocator however can still perform all other real time manual operations such as stop arrived and stop started. The stops are despatched to the driver in the order that they are assigned by Planner Scheduler within the shift. The Allocator can always override this order in which the stops are automatically despatched by setting the stop the Allocator wishes the driver to visit next, to ENROUTE.

5.1.2 Real Time Updates

In the real world, the status of the data objects keeps changing continuously and Planner should be able to reflect these changes and incorporate them into the solution.

The Scheduler comes up with a solution, jobs are then despatched to various drivers and then the drivers carry out their tasks. As vehicles log on each shift, the shift status is updated to STARTED and subsequently to COMPLETED when the vehicles log off. When the vehicle arrives at a stop, the stop status is updated to ARRIVED. When the task that needs to be carried out at the stop is commenced, the stop status changes to STARTED and then to COMPLETED when the vehicle leaves the stop.

Stops, shifts, resources, and jobs can exist with different statusses. These statusses are explained in detail further on in this chapter.

Real time updates to the data objects can be made through-

- PLAN_ packets which advise Planner on the current status of the Data objects.
- Manual Update packets which are requests from the interface to change the status of the data objects.
- MDT updates received from the vehicles that have been delegated to complete the various tasks.

5.1.2.1 Real time Updates through PLAN_ packets

The Data object packets contain static data which is known initially at the time of optimizing the solution. They do not contain any information that would help in determining the current status of the Job, position of the vehicle etc. The PLAN_ packets for Stops, Resources and Shifts supply the additional dynamic information required to describe the current status of each Data object.

- PLAN_RSRC packets advise Planner on the current status of the resource.
- PLAN_SHIFT packets advise Planner on the current status of the shift.
- PLAN_STOP packets advise Planner on the current status of the Stop/Job.

5.1.2.2 Real time Updates through Manual Update packets

These consist of requests from the interface to change the current status of the data object to a new specified status. These requests can be accepted or rejected by Planner. The reason a manual update request is rejected are varied. It may be due to an error in the data packet or due to a conflicting situation that may arise. It may also be a request to change to a status that is not possible at that point of time or under the existing circumstances.

This chapter covers all the Manual Update packets that can be sent from the interface.

5.1.2.3 Real time Updates from the MDT's

Real time update requests to Planner made from the Gateway based on the updates from MDT are explained in detail in the chapter "Mobile Data Messaging" on page 6 - 1.

5.1.2.3.1 Resource Operations

The two manual planning packets for Resources passed on from the Gateway to Planner are RSRC_POSITION and RSRC_DELAY.

RSRC_POSITION - Informs Planner about the new position of the Resource at a specified time.

RSRC_DELAY - Specifies the amount of time by which the Resource will be delayed at any location where it is currently placed.

5.1.2.3.2 Shift Operations

The manual scheduling packets Gateway sends across to Planner to request changes to be made to the statuses of the shift are -

S**HIFT_DISABLE** - Disable the shift so that the shift can no longer be used for planning purposes. All jobs assigned to this shift are reassigned to other shifts or returned to the Free List.

SHIFT_STANDBY - Set the shift-status to STANDBY. The shift will not be used for planning purposes until logged on.

SHIFT_ENABLE - Change the status of the shift from any other status to PLANNED. Planner can assign new jobs to the shift OR reassign jobs from this shift to other shifts.

SHIFT_CLOSE - Set the status of the shift from PLANNED to CLOSED or from STARTED to COMPLETING. Planner cannot make changes to the composition of the shift.

SHIFT_OPEN - Reverse the effects of a SHIFT_CLOSE. Reset the status of the shift from CLOSED to PLANNED or from COMPLETING to STARTED

SHIFT_START - Changes the status of a PLANNED shift to STARTED and a CLOSED shift to STARTING. The shift is assumed to have started at the specified time.

SHIFT_COMPLETE - Changes the status of the shift to COMPLETED. The shift is assumed to have completed all the jobs and the driver is assumed to have logged out.

5.1.2.3.3 Break Operations

Manual scheduling packets from Gateway to Planner requesting a change in the status of Breaks are -

BREAK_DISABLE - The Break is disabled and is not included in the schedule solution.

BREAK_ENABLE - Reset the status of the Break from Disabled. The status of the Break is set to FREE.

BREAK_START - The Break is assumed to have started at the specified time.

BREAK_COMPLETE - The Break is assumed to have been completed. The vehicle is assumed to have departed from the location the break was taken at the specified time.

5.1.2.3.4 Run Operations

RUN_CLOSE -is the manual scheduling packet that Gateway sends across to Planner to request Closing a Run which would prevent any further changes being made to a Run. This operation is applicable to depot based stops only.

5.1.2.3.5 Job Status Operations

Gateway can send Planner the following manual scheduling packets to request simple job status updates -

JOB_DISABLE - The Job is disabled, the stops comprising the Job are not included in the Plan.

JOB_FREE - Set the status of the Job to Free, reset the status of the job to FREE irrespective of it's earlier status.

JOB_ASSIGN - Planner Scheduler assigns the Job to a compatible shift and the status of JOB is changed to ASSIGNED. JOB_ASSIGN can also be used to reset the status of the Job from CLOSED or ALLOCATED.

JOB_ALLOCATE - When a Job is Allocated to a shift, the association of the stops of the specified job/group with the shift can no longer be changed.

JOB_CLOSE - This operation is applicable to depot based stops only. A JOB_CLOSE would request Planner Scheduler to associate the job with a single run (depot stop) and shift. Planner Scheduler cannot change the associated shift or depot stop. **JOB_DESPATCH** - Same as a JOB_CLOSE, in addition to which the driver is notified about the Job.

J**OB_DRIVER_ACK** - Set the status of the Job to ACKNOWLEDGED indicating that the driver has received information on the Job despatch and has acknowledged it.

JOB_DESPATCH_RUN - There may be more than one Job associated with a Run (depot stop). A JOB_DESPATCH _RUN would despatch all jobs associated with the Run.

5.1.2.3.6 Job Assignment Restrictions

The following packets are used to request the Planner Scheduler to place assignment restrictions on Hauls, Depots and Resources -

JOB_FIX_HAUL - Restricts the Job to being associated with any shift within the specified Haul only.

JOB_FIX_DEPOT - Restricts the Job to being associated with the specified Depot stop only.

JOB_FIX_RSRC - Restricts the Job to being associated with any shift that is executed by the specified resource only.

Note: These packets act as toggle operations. JOB_FIX_ can set the restriction or if the restriction already exists, it is UNDONE.

5.1.2.3.7 Stop Status Operations

The following packets are used to send status change requests for single stops to Planner Scheduler -

STOP_DESPATCH - This request is mainly used to undo an earlier ENROUTE, ARRIVE, START, or COMPLETE by resetting the Stop status to DESPATCH or DRIVER_ACK. It should be noted that this operation requests for resetting the status only. It does not actually result in a JOB_DESPATCH message being sent.

STOP_ENROUTE - Places the specified stop as the next stop to be completed in the driver's itinerary.

STOP_ARRIVE - Indicates that the driver has arrived at the Stop at the specified time.

STOP_POSTPONE - Postpones the visit to the stop to occur at a later specified time.

STOP_START - Indicates that the driver has started the load transfer at the stop at the specified time.

STOP_COMPLETE - Indicates that the driver has completed the load transfer at the stop.

5.1.2.3.8 Stop Order Operations

The following manual scheduling packets are used to request restrictions in the order in which the Stop are placed within a schedule -

JOB_INSERT - Inserts a Stop immediately prior to the named Job.

STOP_INSERT - Inserts a Stop immediately prior to the named Stop.

Note: These operations allow for a Job or Stop to be moved within the same haul.

5.1.2.3.9 Group Operations

“Group” is the term used to indicate a set of jobs which for all intends and purposes act as a single job, i.e. manual operations act on all jobs in the group, all jobs must be assigned to the same haul or to the same depot-stop. Most manual operations operate on Groups. However, since an “ungrouped” Job forms a Group in its own right, these operations also act on single, ungrouped, Jobs. Although Groups can be combined into larger Groups, Planner does not recognize levels of grouping. Therefore Ungroup will dissolve groups into Jobs, without any structure.

Group operation requests include -

GROUP - Gathers a list of jobs together to form a Group.

UNGROUP - Removes all jobs from the Group.

UNGROUP_JOBS - Removes the jobs whose Job Idents are specified from the Group.

5.1.2.3.10 Stop Order Restrictions

The following manual scheduling packets are used to request restrictions in the order in which Stops are scheduled -

JOB_LINK - Request to Planner to no longer insert other stops within the Group.

JOB_SEQUENCE - Request to Planner to no longer change the order of stops within the group.

JOB_UNLINK - Request to Planner to undo the effects of a JOB_LINK operation.

JOB_UNSEQ - Request to Planner to undo the effects of a JOB_SEQUENCE operation.

5.1.3 Resource status representation of real world events

The status of the vehicle should represent it’s “Current Position” and it’s “Operational state”.

For the purpose of Planner, a vehicle is at a specific location at a specific time. All timing projections for the route (i.e arrival time at stops) of a vehicle are calculated from its “current position”. During a real time operation the current position of a vehicle will be represented by its last known **position and time**.

The operational state provides information about the vehicle’s availability to perform tasks.

The status of the vehicle is dependent on the status of the shift and the stops assigned to it.

OFF - The vehicle is logged off. It is expected to log on to the next not started shift.

ON -The vehicle is logged on to a shift. It is expected to perform as yet uncompleted tasks allocated to its shift as scheduled.

ARRIVED - The vehicle has arrived at a stop and is waiting to start the stop.

ONSTOP - The vehicle is stopped while the driver is completing one or more stops.

ONBREAK - The vehicle is on one of its scheduled breaks. It is expected to continue its tasks at the scheduled end of the break.

DELAYED - The vehicle is stopped, but expected to resume work shortly.

5.1.4 Shift status representation of real world events

Shifts can have one of the following statusses -

PLANNED - The shift is currently not logged on to, and never has been logged on to.

CLOSED - Same as PLANNED, but will not be recommended for additional work.

STARTED - The vehicle is currently logged on to the shift.

COMPLETING - Same as STARTED, but may not be recommended for additional work.

COMPLETED - A vehicle has logged off from the shift, after completing all jobs.

DISABLED - The shift is not available, no work can be recommended to the shift.

STANDBY - The shift will not be used for planning purposes until logged on.

5.1.5 Break status representation of real world events

Breaks are associated with a shift. Its statusses are -

FREE - The Break is defined, but not expected to be taken.

ALLOCATED - The Break is expected to be taken at some point of time.

STARTED - The Break has been started, i.e the time the break was started is known, with the vehicle assumed to remain at the same location until the break has been completed.

COMPLETED - The break has been completed. The vehicle is assumed to have departed from the location the break was started at the specified time.

5.1.6 Run status representation of real world events

A run consists of a number of stops associated with a single depot stop which can be a distribution or a collection depot stop. Runs can have one of the following statusses -

PLANNED - No restrictions are placed on runs with PLANNED status. Stops can be added, removed or reassigned to other shifts.

CLOSED - A Closed Run consists of a number of allocated Stops and a Depot stop. Planner may not recommend any other stops to be added to or removed from the run. This restriction does not however preclude recommendations from Planner that stops be inserted in between stops belonging to a CLOSED run.

STARTED - The Run has been started, one or more of the stops have been started. The remainder of the run can still be changed for Collection runs. Distribution runs cannot be changed once they have been started.

COMPLETING - The Run has been started - one or more stops (possibly including the depot stop) of the run have been started. The remainder of the run is considered CLOSED.

COMPLETED - All stops (including the depot stop) of the run have been completed.

5.1.7 Job status representation of real world events

Planner does not have separate “job” objects. Hence the Job derives its status from the comprising stops. The PLAN_STOP packets for all stops include a JOB_STATE field which indicates the status of the Job.

The Job statusses are -

FREE - The job has not been assigned yet, no recommendations have been made.

ASSIGNED - Planner has made a recommendation for the possible shift/haul to which the job can be assigned. The job has been tentatively incorporated into a possible schedule. This recommendation can be overridden by the Interface or Planner

ALLOCATED - The job is associated with a particular shift/haul. Planner cannot change this association.

CLOSED -Applicable to depot based stops only. The job is associated with a single run (depot stop) and shift/haul. Planner cannot change this assignment.

PENDING_DESPATCHED- Applicable only to multi-stop Jobs. Similar to ALLOCATED: A Stop is PENDING_DESPATCHED when it is not scheduled within the Despatch Horizon, and another Stop belonging to the same Job is DESPATCHED or higher. Generally PENDING_DESPATCHED Stops maintain this status until they enter the Despatch Horizon and then they change status to DESPATCHED.

DESPATCHED - Same as ALLOCATED, but additionally the driver has been informed about the job.

UNDESPATCHED -The status of a Job which was previously despatched and subjected to Sequence Locking, but is now unlocked due to an out-of-sequence event. Planner treats this Stop as PENDING_DESPATCHED until its status is changed.

ACKNOWLEDGED-The driver has acknowledged receipt of the DESPATCHED job.

STARTED - One stop of the job has been started. A distribution job is also considered STARTED when the depot Stop has been STARTED. Depot Stop is where the goods for delivery are loaded.

COMPLETING - At least one stop of the Job has been completed. A distribution job is also considered COMPLETING when the depot Stop has been COMPLETED. Depot Stop is where the goods for delivery are loaded.

COMPLETED - All stops of the job have been completed.

Note: All Stops of a Job must be on the same haul.

5.1.8 Stop status representation of real world events

Planner interprets a Stop’s status by it’s type. For example Log-Stops are only, either DESPATCHED or COMPLETED. Jobs or Stops that are not associated with depot Stops cannot be CLOSED. For a detailed listing of what all statuses are associated with which stop types see “7.11.10.1 Stop statuses associated with different Stop types” on page 7 - 108.

A Stop can exist in any one of the following statusses -

INACTIVE -The Stop is disabled and cannot be used for scheduling purpose.

FREE - The Stop has not been assigned to any Shift as yet. A single stop can be treated as a one stop Job, in which case, a FREE status means the stop belongs to a FREE job.

ASSIGNED - Planner has made recommendations that associate the Stop to a particular Shift. Planner can reassign the Stop to other shifts as it continues to optimize the solution. The stop belongs to an ASSIGNED job.

ALLOCATED - The association of the Stop has been fixed with a particular Shift. Planner cannot make changes to this Stop/Shift association. The stop belongs to an ALLOCATED job.

CLOSED - This status is applicable to depot based stops only. It is the same as ALLOCATED, and in addition Planner cannot make changes to the current Stop/Run (depot stop) association.

PENDING_DESPATCHED- Applicable only to multi-stop Jobs. Similar to ALLOCATED: A Stop is PENDING_DESPATCHED when it is not scheduled within the Despatch Horizon, and another Stop belonging to the same Job is DESPATCHED or higher. Generally PENDING_DESPATCHED

Stops maintain this status until they enter the Despatch Horizon and then they change status to DESPATCHED.

DESPATCHED - Same as ALLOCATED, and additionally the driver has been advised of the Stop.

UNDESPATCHED -The status of a Job which was previously despatched and subjected to Sequence Locking, but is now unlocked due to an out-of-sequence event. Planner treats this Stop as PENDING_DESPATCHED until its status is changed.

ACKNOWLEDGED-The driver has acknowledged receipt of the DESPATCHED Job.

POSTPONED - The task could not be completed. The driver is assumed to continue to the next scheduled stop, and return to this stop later, but before the end of the shift.

ENROUTE - Same as ALLOCATED, with the additional restriction that the stop will be "done" before any other "ASSIGNED" or "ALLOCATED" stops. The vehicle cannot be diverted until the stop has been completed. This status is used for jobs where it is not practical to inform the driver about the changes to his itinerary, or for the driver to carry out such changes, until this stop has been completed. For example: The vehicle may be out of radio-range, or the driver may not switch on the radio until setting off for the first job of the day.

ARRIVED - The ALLOCATED vehicle has arrived at the location of the stop, but the stop has not yet been started. The vehicle is assumed to remain at the location of the stop until the stop has been completed.

STARTED - The stop has been started. The time the stop was started is known, and the vehicle is assumed to remain at the location of the stop until the stop has been completed.

COMPLETED - The stop has been completed. The vehicle is assumed to have departed from the stop location at the specified time.

Chapter 6: Mobile Data Messaging

This chapter describes the Data Packets and the protocol required for communication between the Planner Client and the MDT (Mobile Data Terminal) via the user-written Gateway software.

6.1 INTRODUCTION

There often arise circumstances when the Allocator sitting in front of the Planner Client interface needs to communicate with the drivers in the vehicles working different shifts. They may want to send the drivers simple, straightforward text messages giving them further instructions about a job/stop etc. They may also want to despatch a Stop or a Job to a particular driver. The drivers will also need to communicate with the Allocators informing them about the job status or delays in a job delivery due to some inevitable circumstances.

Planner can provide support for interaction with drivers through Mobile Data Terminals or MDT's in short. Planner must be able to send job details and miscellaneous messages to drivers, and keep the despatch-Allocator informed about transmission progress and/or failure. Additionally Planner must receive and interpret job/vehicle status update messages and driver to Allocator messages.

Note: Please see "Mobile Data Messages" on page 7 - 13 for the formats of each MDT packet discussed in this chapter.

6.2 PACKETS INVOLVED IN MDT MESSAGING

The data packets used for mobile data messaging are:

- text messages sent from the Planner to the MDT,
- despatch messages sent from Planner to the MDT containing job information about a job that needs to be completed.
- the job status messages sent from the MDT back to the gateway.

These message communications can be viewed by all Planner Clients on the Message Monitor windows. More details on the Message Monitor window are specified in the accompanying Planner User Manual.

6.2.1 Text Messages sent from Client Interface to MDT

When the Allocator sends a message to the driver owning the MDT - the sequence of operations performed are summarized in the table below-

MDT	¤	GateWay	¤	Scheduler	¤	Client
Receives message. Displays to the driver.	‹	Receives message with MESSAGE_ID. Formats the message for the MDT and send to MDT. Hardware acknowledgment of the success or failure of the message transmission to the MDT.	‹	Receives a REQ_TEXT_TO_MDT message from the Client. Generates and attaches a MESSAGE_ID. Sends TEXT_TO_MDT to all Clients and the gateway	‹	Sends the REQ_TEXT_TO_MDT message to Scheduler
					?	Receives TEXT_TO_MDT message. Displays to Allocator.
			?	Receives transmission progress. Sends to All Clients.	?	Receives transmission progress. Displays to the Allocator.

The REQ_TEXT_TO_MDT message initiated by the Allocator is sent across to Planner which adds a Message ID to it and sends a TEXT_TO_MDT across to the gateway and all the Clients. The gateway is responsible for formatting and tailoring the message to the required format needed for transmission to the MDT. The gateway receives a hardware acknowledgment of the success or failure of this operation which is then, sent across to the Server and Client interface.

6.2.2 Despatching jobs from the Client interface to the MDT

A single stop or a group of stops (Job) can be despatched across to the driver. The sequence of operations is as shown below-

MDT	¤	GateWay	¤	Scheduler	¤	Client
Receives message. Displays to the driver.	‹	Receives the JOB_TO_MDT packet. Formats the message for the MDT and sends to MDT. Reports on transmission progress.	‹	Receives the DESPATCH request. Generates a JOB_TO_MDT packet and attaches a MESSAGE_ID. Sends to All Clients.	‹	Sends a request REQ_DESPATCH to Scheduler
					?	Receives JOB_TO_MDT packet. Displays to Allocator.
			?	Receives transmission progress. Sends to all Clients.	?	Receives transmission progress. Displays to Allocator.

The Allocator at the Client interface initially requests a Job despatch - REQ_DESPATCH to be sent across to the driver. Scheduler adds the required information or the specified information (the job specifications that need to be sent across to the MDT can be specified in the config file) and generates a JOB_TO_MDT packet. This packet gets sent across to all the Planner Client interfaces and the gateway.

The gateway will then tailor it into a format needed to transmit across to the MDT and send it across to the MDT. The JOB_TO_MDT packets caters for jobs of up to 2 stops only. For jobs with a larger number of stops multiple individual messages must be sent.

When AutoDirect is activated, the manual despatch request functionality is disabled. That is, Manual Despatch will not send the JOB_TO_MDT message to the driver. For more information on AutoDirect, see “Real Time Customization” on page 5 - 2

6.2.3 Message transmission updates from Gateway to Planner

The MDT on receipt of a message sends back a reply to the gateway. Based on this reply, the Gateway reports the progress of the message to the Client and Planner with a MSG_STATUS packet.

This packet defines the following states for the messages -

IN_PROGRESS - the message is being transmitted to the MDT

RETRYING - the transmission of the message to the MDT has not been successful and the Gateway is retrying to send the message to the MDT

FAILED - the transmission of the message to the MDT has failed.

SUCCEEDED - the transmission of the message was successful

6.2.4 Status updates

Status update messages are sent from Gateway to Planner based on the feedback obtained from the drivers in the vehicles. Gateway sends packets to Planner requesting Planner to change the status of the vehicles, shifts and stops accordingly.

MDT_ARRIVED, MDT_STARTED, MDT_COMPLETED notify the arrival, starting and completion of a task at a stop. Other Job update packets include MDT_ENROUTE, MDT_ACK and MDT_DEFERRED.

MDT_ENROUTE specifies that the vehicle will advance towards the specified stop. MDT_ACK acknowledges receipt of the specified job. MDT_CLRSTAT indicates that the driver wishes to revert back to the default status - ACKNOWLEDGED for Job stops and ALLOCATED for depot stops. MDT_DEFERRED indicates that the driver wishes to postpone the visit to a particular stop to a later time within the duration of the shift.

Vehicle position and status updates are passed on through the following packets - MDT_CURPOS, MDT_LOGON, MDT_BREAK, MDT_DELAYED, MDT_LOGOFF.

Chapter 7: Packets (10.5)

This chapter describes the interface packet changes in Oracle Real-Time Scheduler 10.5 since Oracle Real-Time Scheduler 10.4, functionalities supported by Oracle Real-Time Scheduler packets, the system limits for Oracle Real-Time Scheduler Objects, the events handled by Oracle Real-Time Scheduler, the Oracle Real-Time Scheduler data definitions and the packet formats.

7.1 CHANGES

7.1.1 Packet changes from 10.4 to 10.5

7.1.1.1 New packets in Oracle Real-Time Scheduler 10.5

Packet	Page	Description
SR_BATCH_ON	Page 2	Sentinal elements to indicate start and end of Appointment Booking Request packets
SR_BATCH_OFF	Page 2	
SR_BATCH_DONE	Page 2	Packet to indicate that the Appointment Booking Request has been received and processed
CA_BATCH_ON	Page 2	Sentinal elements to indicate start and end of Conditional Assignment Request packets
CA_BATCH_OFF	Page 2	
CA_BATCH_DONE	Page 2	Packet to indicate that the Conditional Assignment Request has been received and processed
SR_CHS_SLOT	Page 138	The Appointment Booking Request Packet introduced in Planner 10.5 to extend and eventually replace the existing CHS_SLOT packet functionality
CA_JOB_ASSIGN_COND	Page 126	The Conditional Assignment Request Packet introduced in Planner 10.5 to extend and eventually replace the existing JOB_ASSIGN_COND packet functionality

7.1.1.2 Modified 10.4 Packets in Oracle Real-Time Scheduler 10.5

FIELD	Page	Description
STOP	Page 60	One spare field replaced with new field: FIXED_RSRC

7.1.1.3 Packets from 10.4 that are not in Oracle Real-Time Scheduler 10.5

None

7.1.2 Packet changes from 10.3 to 10.4

7.1.2.1 New packets in Oracle Real-Time Scheduler 10.4

None.

7.1.2.2 Modified 10.3 Packets in Oracle Real-Time Scheduler 10.4

FIELD	Page	Description
PLAN_STOP	Page 101	One spare field replaced with new field: BIND_MODE
STOP	Page 60	Two spare fields replaced with new fields: R_SKILLS and R_CREW_SIZE
SHIFT	Page 78	One spare field replaced with new field: CREW_SIZE ZONE_PREF field's default changed to ALL from NONE RSRC_ATTRIBUTES field replaced with SKILLS

7.1.2.3 Packets from 10.3 that are not in Oracle Real-Time Scheduler 10.4

None

7.1.3 Packet changes from 10.2 to 10.3

7.1.3.1 New packets in Oracle Real-Time Scheduler 10.3

None.

7.1.3.2 Modified 10.2 Packets in Oracle Real-Time Scheduler 10.3

FIELD	Page	Description
CLIENT_TYPE	Page 18	VERSION field value modified from (6_0 - 10_0) to (6_0 - 10_3)
STOP	Page 60	One spare field replaced with new field: BIND_MODE
SHIFT	Page 78	Two spare fields replaced with new fields: SEQ_LOCKING and AUTO_UNDESP

7.1.3.3 Packets from 10.2 that are not in Oracle Real-Time Scheduler 10.3

PACKET	Description
PLAN_INIT	The functionality for which this packet was intended was never implemented
INSERT	This packet has never been used and although implemented differently it duplicates the INSERT_JOB and INSERT_STOP packets
JOB_FIX_DEPOT	The functionality for which this packet was intended was never implemented

7.2 FUNCTIONALITY

The packet definitions must support the following functionality:

7.2.1 Process identification:

- Each process within Oracle Real-Time Scheduler must be advised of the type of a connected process, the host it runs on and its “pid”.

7.2.2 Real Object identification:

- Packets identify real objects with an id which is unique within the class. (DEPOT, RESOURCE, SHIFT, STOP, TIME-WINDOW, SPEED-WINDOW, BREAK)

7.2.3 Virtual Object identification:

- Packets identify virtual objects with an id which is unique within the class. (HAUL, RUN, GROUP)

7.2.4 Multiple Clients:

- SmAuto acts as a single repository of Planner’s “world”. It services a number of GUIs, each of which may have a different “view” or filter to select the data relevant for the specific user.
- Scheduler will provide default communications parameters for certain types of Client.
- Any Client can independently customize its communication parameters.

7.2.5 Real Time Planning:

- SmAuto immediately incorporates new data into its current schedule.
- New data will be accepted and processed at any time.

7.2.6 Manual Planning:

- All manual planning is achieved through special purpose packets.
- Jobs can be grouped to simplify manipulation.
- Manual assignments cannot be undone by Scheduler.

7.2.7 Scheduler minimizes:

- Total distance travelled.
- Total working time.
- Arrival after the end of a specified time-window.
- Number of Shifts used.
- Number of Resources used.
- Number of Stops in non-preferred zones.
- The use of undesirable Shifts.
- The use of undesirable time-windows.
- Overlap in runs for the same Resource.

7.2.8 Scheduler recognizes that:

- Stops may have multiple time-windows.
- Depots may have multiple time-windows.
- Successive Shifts may be combined into a single Haul.
- Goods may be obtained from/delivered to more than one Depot.
- Visiting a site once to do more than one Job may reduce the total time spent on this site.
- Delay-time at the Depot may depend on the total amount of goods to be (un)loaded.
- Average speeds for un-mapped areas may vary.
- Resources may be restricted to or prefer certain zones.
- Some types of goods cannot be carried concurrently.
- Some types of goods require other types of goods to be carried concurrently.
- Some types of goods cannot be carried by specific Resources.
- Some types of goods must be carried without intervening Stops.
- Some types of goods must be carried separately.
- Some types of goods have a maximum time between Collection and Distribution.

7.3 SYSTEM LIMITS

Most objects have a fixed upper limit on the population size within SmAuto. When a packet received by SmAuto causes any of these limits to be exceeded, it will be rejected.

Note: These limits apply any time, so also during transitional states bracketed by BATCH_ON, BATCH_OFF. Temporary violations are not permitted.

System Limits (Sheet 1 of 2)

Object	Limit	Description
Shifts	16000	The maximum total number of Shifts that Scheduler can contain.
Resources	16000	The maximum total number of Resources that Scheduler can contain.
Resource Shifts	128	The maximum number of Shifts that can be assigned to a Resource.
Haul Shifts	8	The maximum number of Shifts that can be assigned to Shifts comprising a shift.
Haul IDB	263	Within the plan we are limited to 256 stops per haul. Because any logoff/logon is represented by a single "HaulStop" (aka "RestStop"), in the IDB, this is the maximum stops per haul.
Non-Log Haul	254	A haul has at least 2 log-stops, therefore the number of non-log stops cannot exceed 256 minus 2.
Stops	60000	The maximum total number of Stops that Scheduler can contain. **Note:** Each Shift in the plan uses up two additional Stops, and each run one.
Stops per Shift	256	Maximum number of scheduled stops for a shift (with defined position in a shift)
Stops per Job	254	Derived as equal to ("Stops per Shift" - 2) Maximum number of stops that can be associated with a single Job
Depots	1000	The maximum total number of Depot Stops that Scheduler can contain
Zones	1024	The maximum number of unique zone-identifiers.
Products	1000	The maximum number of unique product identifiers used to describe products delivered to or collected from Depots.
SpeedControl Granularity	900	Maximum number of SpeedControl granularity that can exist in the system. SpeedControl granularity = 15 minutes
Dated Speed Profiles	120	Maximum date type speed time window profiles that can exist in the system.

System Limits (Continued) (Sheet 2 of 2)

Object	Limit	Description
Speed Profiles	128	Derived as (1 + 7 + Dated Speed Profiles) Maximum speed time window profiles that can exist in the system. This is the sum of a single default profile, seven day of week profiles and the date type speed profiles defined using the Dated Speed Profile limit.
Max Time Windows Per Stop	128	Maximum number of time windows for a single Stop.
Max Time Windows Per Depot Stop	128	Maximum number of time windows for a single Depot.
Max Stop Time Windows in System	300000	Maximum number of Stop time windows that Scheduler can handle at any time
Max Depot Time Windows in System	128000	Derived as equal to ("Depots" x "Max Time Windows Per Depot Stop") Maximum number of Depot time windows that Scheduler can handle at any time
Depots/Job	12	Maximum number of Depots that a Job may be sourced from/ delivered to
View	256	The maximum number of unique view-identifiers.
Job Attribute	128	The maximum number of unique identifiers for Job to Job compatibility attributes
Job Attribute Conflicts	16	Maximum number of POTENTIAL Job-attribute in a TOUR.
Breaks	8	Maximum number of Breaks that could be taken during a Shift
Timeslots	2000	Maximum number of slots to be used in calculating site separation cost (48 half-hours x 31 days)
Slot Groups	1000	Derived as equal to ’Depots’ in the system. Maximum number of slot-groups. Groups are expected to be used to provide groups of slots per depot.
Slots per Group	(28*24)=672	Maximum slots within each group to be able to handle 28 days of 24 slots each.
Max Slots	672000	Derived as equal to ("Slot Groups" x "Slots per Group") Maximum slots that Scheduler can hold at a time. Calculated as (24 slots per day for 28 days for 1000 Depots).
Max Secondaries	16	Maximum number of secondary stops in a Bound job.
Max Bond	17	Derived as equal to (Max Secondaries + 1) Maximum number of stops in a bound Job.
Identifier	Identifiers must not contain leading or trailing white-space.	

7.4 EVENTS

This section details a list of all conceptually atomic events that Oracle Real-Time Scheduler handles.

7.4.1 Load/Save

Once-off loading or saving data initiated by an Allocator.

Event	Description
Load data from HOST	Oracle Real-Time Scheduler selects and retrieves data from the HOST and updates its own tables.
Save data to HOST	Oracle Real-Time Scheduler selects data from its own tables and updates the HOST ***with allocation info only***.
Load data from text-file	Oracle Real-Time Scheduler loads data from a file in text format.
Save data to text-file	Oracle Real-Time Scheduler writes selected data to a text file.

7.4.2 Object Edit

The Allocator modifies the description of a Stop, Resource etc. The objective is to modify the properties of the object, not specifically to modify its position in the current plan. The change is broadcast to all Clients.

Event	Description
Edit a Depot	The Allocator modifies the description of a Depot within Oracle Real-Time Scheduler.
Edit a Resource	The Allocator modifies the description of a Resource or Resource within Oracle Real-Time Scheduler.
Edit a Shift	The Allocator modifies the description of a Shift within Oracle Real-Time Scheduler.
Edit a Stop	The Allocator modifies the description of a Stop within Oracle Real-Time Scheduler.
Edit a Break	The Allocator modifies the description of a Break within Oracle Real-Time Scheduler.

7.4.3 Plan Edit

The Allocator modifies the plan, e.g. the Shift a Stop or Job is allocated to. The objective is to change the plan, not object properties. In order to prevent conflicts, this functionality is provided by SmAuto, and accessed by the Clients through request messages.

Note: If a Job is part of a group, the specified operation applies to the group, not just the Job.

Event	Description
Group Jobs	A selected set of Jobs is to all intents and purposes treated as a single Job. Any “group” operation applied to a single member of a group will affect all members simultaneously.
Add a single Job to a Group	
Group a Run	Groups all Job serviced by a single Depotstop. The Depotstop is included in the Group. The Depotstop services all and just members of the group.
Dissolve a Group	These Jobs are now no longer linked together in a larger unit. Each of the previous members of the “group” is now an individual independent Job. Any “group” operations applied to a Job that has been removed from the group will affect that Job only, rather than all Jobs in the group.
Remove a single Job from a Group	
Allocate a Job to a Resource	These assignments cannot be undone by Scheduler.
Allocate a Job to a Haul	
Allocate a Job to a Shift	
Un-Allocate a Job	
Link	All Stops have to be executed consecutively. (as a. Single Link)
UnLink	
Sequence	All Stops have to be executed in the prescribed order.
UnSequence	
Close a Shift	Stops may only be allocated or despatched to this Shift. No Stops may be assigned to the Shift. **Note: Although a single Job may have Stops allocated to more than one Shift (within the same haul), only those Stops that are allocated to the specified Shift are affected.**
UnClose a Shift	
Position a Stop	Move a single Stop to a specific position in the schedule.
Optimize a Job	Oracle Real-Time Scheduler re-optimizes a haul from the first Stop to the last of the set of Jobs indicated.
Optimize a Shift	
Optimize a Haul	

7.4.4 General Updates

Unsolicited general updates to a Stop, Resource etc. received from the Host.

Event	Description
Object update	Oracle Real-Time Scheduler receives an update originating in the Host.

7.4.5 Real-Time Updates

Unsolicited updates specific to Real-Time operation.

Event	Description
Update Resource position	Oracle Real-Time Scheduler receives a new Resource position.
A Stop has been started	Oracle Real-Time Scheduler is advised a specific Stop has been started/completed.
A Stop has been completed	
A Stop is returned to “allocated”	A Stop that was previously (thought to be) started/completed is reset to “allocated”.

7.4.6 Recommendation Requests

The Allocator requests a recommendation for the allocation of a Job or Group, without actually affecting the current plan.

Note: If a Job is part of a group, the specified operation applies to the group, not just the Job!~~.~~

Event	Description
Recommend Shift	Oracle Real-Time Scheduler finds the best candidate Shifts for a Job.
Recommend Slot	Oracle Real-Time Scheduler finds the best candidate time Slot for a Job to be include in plan

7.4.7 Control Edit

The Allocator modifies a parameter controlling the behavior of Scheduler.

Event	Description
Edit CostControl	Modifies the relative weights of the various factors that contribute to the objective function of Scheduler. Changes Scheduler behavior.
Edit SpeedControl	Modifies the assumed speed of Resources depending on time-of-day.

7.4.8 System commands

Specific commands to start or Stop Scheduler, switch to Real-Time etc.

Event	Description
Reset Oracle Real-Time Scheduler	Removes all data and definitions.
Activate Scheduler	Scheduler will attempt to improve on the current solution.
Deactivate Scheduler	
Switch Oracle Real-Time Scheduler to Real-Time planning	When in Real-Time mode, the Oracle Real-Time Scheduler will interpolate and adjust Resource positions.
Switch the Oracle Real-Time Scheduler to Forward planning	

7.4.9 Client status

Changes to the status of a Client.

Event	Description
A Client attaches to SmAuto	A new Client-process is started. As it connects it transmits identification and other details.
A Client detaches from SmAuto	The Client-process advises SmAuto and disconnects.
SmAuto dies	Client detect that SmAuto has disconnected.
A Client's view changes	A Client requests that SmAuto changes the range of data it sends.

7.4.10 SmAuto status

Changes to the status of SmAuto.

Event	Description
A Client dies	SmAuto detects that a Client-process is no longer connected.
Scheduler changes the mode it operates in	SmAuto changes one of its operational modes: Real-Time, Optimize on/off.
Scheduler progress	Scheduler status and progress.
Scheduler in server or client mode	Scheduler starts up as a Client requiring connection from Oracle Real-Time Scheduler Switch process, or as Server process acting independent of Switch process.

7.4.11 Mobile Data Messages

Messages to/from MDT.

Event	Description
A Client requests SmAuto to send a message to an MDT	SmAuto attaches a message-id and passes the message on to GateWay.
A Client requests SmAuto to despatch a Job to an MDT	SmAuto generates a message and passes it on to the GateWay.

7.4.12 Miscellaneous

Event	Description
Inter-Client messages	Messages between Clients.

7.5 DATA DEFINITIONS

The data which is required for Oracle Real-Time Scheduler is grouped by functionality, and is described in tables according to the grouping. The data definitions in this chapter are independent of the origin of the data itself or the method used to communicate it to Oracle Real-Time Scheduler.

7.5.1 Syntax

7.5.1.1 Table Contents

The tables that describe the data required for Oracle Real-Time Scheduler contain the following information about every data element:

1. Field Name- the name by which the field is known to Oracle Real-Time Scheduler.
2. M (Mandatory)- an entry in this table column means that the data is mandatory. No default value is provided for mandatory data, and its absence will cause the data to be rejected. (See 5.Range)
3. Format- the nature of the data (see Terminology below).
4. Description- shows what the data means to Oracle Real-Time Scheduler.
5. Range- the range of data values which are acceptable to Oracle Real-Time Scheduler. For some non-mandatory data fields, a default value will be shown in square brackets []. The default value will be used by Oracle Real-Time Scheduler if this data field is empty.

Note: The “Range” column will display the data limits according to the Format specified for the data. For example, if the data format is TOKEN then all the acceptable TOKENS will be shown.

7.5.1.2 Terminology

Term	Meaning
CARDINAL	A cardinal number, that is, a positive integer - e.g., 0,1,2,3, etc.
INTEGER	An integer number, either positive or negative - e.g., -2, -1, 0, 1, 2 etc.
DOUBLE	A decimal number, either positive or negative - e.g., 97.89, -3124, 3.14159....
STRING	A continuous sequence of printable ASCII characters, no NULL characters.
IDENT	As for STRING, but does not allow leading or trailing whitespace characters.
TOKEN	One of a number of pre-defined STRINGS - e.g., DEPOT
DATE	A Julian calendar date, expressed as yyyymmdd - e.g., 19990426
HHMM	A "time of day" or "period of time" in 24 hour format, expressed as hhmm - e.g., 1442

Term	Meaning
HHMMSS	A "time of day" or "period of time" in 24 hour format, expressed as hhmmss - e.g., 144203

7.6 PACKET TYPES

Packet descriptions and definitions are grouped by functionality. The packets making up each functional category are detailed in the following sections.

The Oracle Real-Time Scheduler packets have been categorized under the following functional headings:

- Client-Server Interface Packets
- Scheduler Control Packets
- Real Time Packet
- Manual Planning Packets
- Object Packets
- Object Delete Packets
- Plan Packets
- Plan-Parameter Packets
- Status Messages Packets
- Mobile Data Terminal Text Message Packets
- Print Manifest Server Interface Packets
- Switch Interface Packets

7.7 CLIENT - SERVER INTERFACE PACKETS

The packets discussed in this section define the interface between Oracle Real-Time Scheduler Client and SmAuto.

Client - Server interface Packets

Packet Type	Description
CLIENT_TYPE	Identifies a Client with a “type” and a specific id.
CLIENT_CONNECT	Client requests SmAuto for connection either directly or via GUI Server.
CLIENT_REJECT	GUI Server rejects the Client connection request.
CONNECT	A Client wishes to connect/has connected to SmAuto.
DISCONNECT	A Client wishes to disconnect/has disconnected from SmAuto.
USER_AUTH_REQ	Scheduler requests User/Password from Client to determine user’s access level.
USER_CONNECT	Client passes the User/Password obtained from the user to Scheduler in the USER_CONNECT packet
AUTH_RESPONSE	Scheduler sends the appropriate Access Level to the connected Client.
AUTH_ALLOW	Scheduler sends an allowed operation message to plannerClient.
AUTH_DENY	Scheduler sends a denied operation message to plannerClient.
REPORT_ON	Requests SmAuto to send object updates when they occur. Initially to update the Client with the current state, and subsequently whenever there is a change in SmAuto.
REPORT_OFF	Requests SmAuto to maintain a list of objects for which the Client needs an update, but not to synchronize the Client until REPORT_ON has been received.
AUTH_COMPLETE	Scheduler sends an AUTH_COMPLETE to the Client to terminate a USER_AUTH_REQ.
VIEW	Specifies the range of objects the Client wishes to access.
VIEW_ALL	Specifies the range(s) of objects the Client is able to access.

7.7.1 Client Connection

When a Client connects to SmAuto, SmAuto will take no action other then to wait for a CLIENT_TYPE packet. This packet informs SmAuto about the kind of Client is connected (GUI, GATEWAY) and which version of the interface. If VERSION is “4.2”, SmAuto will communicate using the packets described in Chapter 7: “Packets (10.5)” on page 7 - 1. Until the CLIENT_TYPE -packet has been received, no other packets will be processed, and no data or messages will be sent.

Client Identification Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Identifies the Client Type	CLIENT_TYPE
IDENT	X	IDENT	Unique identifier for the Client.	
VERSION	X	TOKEN	Interface version.	6_0 to 10_3
TYPE	X	TOKEN	Classifies the Client.	GUI, GATEWAY, SCHEDULER, NONE

On receipt of a “CLIENT_TYPE” packet SmAuto must:

- Set default-behavior.
- Receive a CLIENT_CONNECT packet
- Send USER_AUTH_REQ packet if UAC is enabled and expect a USER_CONNECT from the Client in return.
- Send AUTH_RESPONSE packet followed by (AUTH ALLOW/DENY packets based on the granted Access level or AUTH ALLOW_ALL packet if UAC is disabled).
- Terminate the USER_AUTH_REQ by sending an AUTH_COMPLETE packet to the Client.
- Broadcast a CONNECT to all Clients.
- Receive a VIEW packet from the Client.
- Receive a SYNC_PLAN from the Client.
- ~~Return an ALL_VIEW packet containing the list of possible views.~~

Client Connection Request Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Requests Client connection	CLIENT_CONNECT
IDENT	X	IDENT	Unique identifier for the Client.	

PlannerClient sends the CLIENT_CONNECT packet to SmAuto, either directly or through the GUI Server (depending on the host system setup). The GUI Server will check the Client ID supplied within the CLIENT_CONNECT packet and send back a CLIENT_REJECT in the event of having deciphered a duplicate Client ID.

Packet Rejecting Client Connection

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Rejects Client Connection request	CLIENT_REJECT

7.7.2 User Access Control Packets

The User Access Control packets interact between the Client and Scheduler to determine the User's Access Level.

Once User Access Control is activated the Oracle Real-Time Scheduler Client will enable only those user operations that are permitted for the active Access Level. See "Client Operations at the default Access Levels" on page 7 - 25.

7.7.2.1 Client Connects to Scheduler

A Client, opens TCP/IP socket connection to Scheduler and sends a CLIENT_TYPE packet identifying the interface version the Client will be using and a CLIENT_CONNECT packet specifying the Client ID. As Clients have a concept of what their Access Level is they default to having most options disabled (VIEW_ONLY level) until told otherwise by Scheduler.

7.7.2.2 Reading User Access Groups file.

SmAuto reads the User Access Groups file (if specified) only once at startup. Hence after adding or changing a User Access Group, Oracle Real-Time Scheduler needs to be restarted.

Note: Upon start-up if the User Access Groups or User Access Control File cannot be opened (say due to wrong file location, invalid permission or because the file does not exist) Scheduler will halt operation and exit with an appropriate error message

7.7.2.3 Scheduler determines if User Access Control is enabled.

SmAuto reads the User Access Group file at startup and the User Access Control file whenever a Client requests for connection, with User Access Control enabled.

The User Access Group File defines the path and name of the file containing the access groups. This file specifies the operations which would be allowed or denied for each Access Group. A default User Access Groups File consisting of five groups is provided with each Oracle Real-Time Scheduler installation. See 7.7.4 "Default User Access Group File" on page 7 - 23

The User Access Control File defines the path and name of the user/password file the contents of which will be used to authenticate connections. See accompanying Oracle

Real-Time Scheduler "Installation Manual" for further details on User Access Control. By default these configuration parameters are set to NULL, thus disabling User Access Control. Both the parameters must be defined to enable User Access Control.

Note: If User Access Control is disabled then all Clients connecting to Scheduler are granted SYSTEM level access.

7.7.2.4 Scheduler sends Authorisation Request to Client

The Authorisation Request packet will be the first packet sent by Scheduler, after receiving the CLIENT_CONNECT packet with User Access Control enabled. All packets other than USER_CONNECT (see “Client Responds with Username & Password” on page 7 - 20) received from the Client will be rejected by SmAuto with an appropriate error

Authorization Request Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Sent by SmAuto upon receipt of a CLIENT_TYPE packet, requesting Client respond with USER_CONNECT packet.	USER_AUTH_REQ
NEW_PASSWORD		TOKEN	Will be set to YES if the User is required to change their password immediately.	YES,[NO]
spare			May be used to extend functionality	
spare				

7.7.2.5 Client Responds with Username & Password

The Client passes the username and password obtained from the User to Scheduler in the USER_CONNECT packet.

User Verification Request Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Sent by Client in response to a USER_AUTH_REQ	USER_CONNECT
USERNAME	X	STRING	User name of User attempting to establish a connection	
PASSWORD		STRING	Encrypted password string	
NEW_PASSWORD		STRING	Encrypted password string, set when changing Users password.	
spare			May be used to extend functionality	
spare				

7.7.2.6 Client Notification

After the user access group has been determined by SmAuto, SmAuto needs to send all the allowed/denied operations for the user to PlannerClient.

SmAuto first sends the AUTH_RESPONSE packet, and then a series of AUTH_ALLOW and AUTH_DENY packets.

Authorization Response Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Sent by SmAuto upon receipt of a USER_CONNECT packet, granting a certain User Access level.	AUTH_RESPONSE
USERNAME	X	STRING	User name of User attempting to establish a connection	
ACCESS_LEVEL		STRING	User Access level granted to the Client.	
NEW_PASSWORD		DATE	Notifies the User that their current password will expire on this date. This field will only be filled if the expiry date is within the notification period.	
spare			May be used to extend functionality	
spare				

Operations allowed/denied for the Client

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Sends an allow/deny message to plannerClient.	AUTH_ALLOW, AUTH_DENY
ACCESS_GROUP	X	IDENT	Identifies the User Access Group.	
OPERATION	X	TOKEN	Identifies the operation to be allowed or denied.	
spare				
spare				
spare				

Note: SmAuto terminates each authorization request with an AUTH_COMPLETE packet. See AUTH_COMPLETE packet format under section 7.7.8.1 “Reporting Events” on page 7 - 30.

If an unknown User name or invalid password is supplied then Scheduler resends the USER_AUTH_REQ packet to the Client and after a predefined number of invalid access attempts disables the Client with a DISABLED access level

List of Default Access Levels

Token	Description
VIEW_ONLY	At this access level the connected Client would have all data and planning related operations disabled. They could view the data but not modify it in any way. Scheduler would also reject all attempts to modify data.
PLANNING	At this access level the connected Client could make plan requests to Scheduler. These request would include the ability to, move Jobs between Shifts and assign and allocate Jobs within the plan.
REAL_TIME	At this access level the connected Client could make Real Time requests. These would include starting and completing Jobs.
DATA_MOD	This access level would allow data to be modified. Fields data objects (Stops, Shifts etc.) could be modified. New Stops and Shifts could NOT be created.
SYSTEM	At this access level data can be loaded and saved. New objects can be created and modified. Scheduler state (scheduling or idle) could be adjusted and the real time awareness of the system could be modified.
DISABLED	Scheduler sends the AUTH_RESPONSE packet to a User with ACCESS_LEVEL set to DISABLED if the number of invalid password entries for the User exceeds the pre configured (command line argument) limit.

7.7.3 User Access Control File

The following table lists the format for making entries into the User Access Control File. The field separator is a ‘:’.

For example: Michael:42XDSPELDIFNQ:20032906:VIEW_ONLY

Access Control File Format

Field Name	M	Format	Description	Range
USER_NAME	X	STRING	Case insensitive User name of User attempting to establish a connection.	
PASSWORD	X	STRING	Encrypted password of User attempting to establish a connection	
EXP_DATE	X	DATE	Date password expires and User is required to enter a new password.	
ACCESS_LEVEL	X	TOKEN	User Access level granted to the Client	VIEW_ONLY, PLANNING, REAL_TIME, DATA_MOD, SYSTEM Any other USER_DEFINED

7.7.4 Default User Access Group File

Oracle Real-Time Scheduler comes complete with a default access Group file, consisting of five access group definitions. This file controls the operations that will be made available at each of the available access levels.

```
group VIEW_ONLY
{
deny ALL
}

group PLANNING
{
inherit VIEW_ONLY
allow
CLOSE_ALL_SHIFTS,CLOSE_ALL_RUNS,ALLOCATE_ALL_STOPS,RESET_PLAN
allow
SHIFT_DISABLE,SHIFT_ENABLE,SHIFT_CLOSE,SHIFT_OPEN,SHIFT_OPTIMIZ
E
allow
BREAK_DISABLE,BREAK_ENABLE,RUN_CLOSE,RUN_REMOVE,RUN_ALLOCATE
allow
JOB_DISABLE,JOB_FREE,JOB_ASSIGN,JOB_UNALLOCATE,JOB_ALLOCATE,JOB
_ALLOCATE_SHIFT
allow
JOB_CLOSE,JOB_FIX_PLAN_HAUL,JOB_FIX_HAUL,JOB_FIX_PLAN_RSRC,JOB_
FIX_RSRC
allow CHS_SHIFT
}

group REAL_TIME
{
inherit PLANNING
allow RSRC_POSITION,RSRC_DELAY, RSRC_DELAY_COMPLETE
allow
SHIFT_START,SHIFT_COMPLETE,SHIFT_POSITION,SHIFT_DELAY,SHIFT_STA
NDBY,GO_HOME,RESUME_WORK
allow BREAK_START,BREAK_COMPLETE
allow RUN_DESPATCH,RUN_ACK,RUN_ENROUTE
allow JOB_DESPATCH,JOB_DESPATCH_SHIFT,JOB_DRIVER_ACK
allow
STOP_DESPATCH,STOP_ENROUTE,STOP_ARRIVE,STOP_POSTPONE,STOP_START
,STOP_COMPLETE,STOP_POSITION
allow REQ_TEXT_TO_MDT
allow AUTO_DIRECT
}
```

```
group DATA_MOD
{
inherit REAL_TIME
allow SAVE_DATA,SAVE_AS_DATA
allow EDIT_COST_CONTROL,PREF_BEHAVIOUR,EDIT_SPEED_TW_PROFILE
allow EDIT_RSRC,EDIT_SHIFT,EDIT_BREAK,EDIT_DEPOT,EDIT_DEPOT_TW
allow EDIT_DEPOT_PROD,EDIT_STOP,EDIT_STOP_TW
allow GROUP,UNGROUP_JOBS,UNGROUP,JOB_LINK,JOB_UNLINK,JOB_SEQUENCE,JOB_UNSEQ
allow JOB_TRANSFER,BIND_STOP,UNBIND_STOP,TIGHTEN_BOND,SET_SUBWINDOW
}

group SYSTEM
{
inherit DATA_MOD
allow LOAD_DATA,LOAD_GATEWAY,MERGE_DATA,SAVE_GATEWAY,RESET
allow SCHED_ON,SCHED_OFF
allow HAUL_DEL,RSRC_DEL,SHIFT_DEL,BREAK_DEL,DEPOT_DEL,JOB_DEL,STOP_DEL
allow DEPOT_CUTOFF
allow REALTIME,USER_COMMAND

# alternatively the allow all option can be used for SYSTEM
# allow ALL
}
```

7.7.5 Client Operations at the default Access Levels

Upon receiving an Authorisation Response packet the Oracle Real-Time Scheduler Client will disable the menu items and operations no longer available to the User.

Operations such as Colour Rules and window layout, which only effect the Oracle Real-Time Scheduler Client, are independent of User Access Control

Client Operations at the default Access Levels (Sheet 1 of 4)

Operations	VIEW_ ONLY	PLANNING	REAL_ TIME	DATA_ MOD	SYSTEM
Load HIP					X
Merge HIP					X
Load from Gateway					X
Save HIP					X
Save HIP Version					X
Save to Gateway					X
Reset					X
Real Time Mode			X	X	X
View	X	X	X	X	X
Update	X	X	X	X	X
Start Optimiser					X
Stop Optimiser					X
Select All Stops		X	X	X	X
Select All Shifts		X	X	X	X
Close All Shifts		X	X	X	X
Close All Runs		X	X	X	X
Allocate All Stops		X	X	X	X
Reset Plan		X	X	X	X
View Resource	X	X	X	X	X
Edit Resource				X	X
Reposition Resource			X	X	X
Delay Resource			X	X	X
Delete Resource					X
Delete Haul					X
View Shift	X	X	X	X	X
Edit Shift				X	X
Enable Shift		X	X	X	X

Client Operations at the default Access Levels (Continued) (Sheet 2 of 4)

Operations	VIEW_ ONLY	PLANNING	REAL_ TIME	DATA_ MOD	SYSTEM
Disable Shift		X	X	X	X
Standby Shift			X	X	X
Start Shift			X	X	X
Complete Shift			X	X	X
Close Shift		X	X	X	X
Open Shift		X	X	X	X
Optimise Shift	X		X	X	X
Delete Shift					X
View Break	X	X	X	X	X
Edit Break				X	X
Disable Break		X	X	X	X
Enable Break		X	X	X	X
Start Break			X	X	X
Complete Break			X	X	X
Delete Break					X
View Depot	X	X	X	X	X
Edit Depot				X	X
View Depot Time Window	X	X	X	X	X
Edit Depot Time Window				X	X
View Product Depot	X	X	X	X	X
Edit Product Depot				X	X
Delete Depot					X
Close Run		X	X	X	X
Remove Run		X	X	X	X
Allocate Run		X	X	X	X
Despatch Run			X	X	X
Acknowledge Run			X	X	X
Run Enroute			X	X	X
Print Run		X	X	X	X
Group Jobs		X	X	X	X
Ungroup Jobs		X	X	X	X
Ungroup		X	X	X	X

Client Operations at the default Access Levels (Continued) (Sheet 3 of 4)

Operations	VIEW_ ONLY	PLANNING	REAL_ TIME	DATA_ MOD	SYSTEM
Link Job		X	X	X	X
Unlink Job		X	X	X	X
Sequence Job		X	X	X	X
Unsequence Job		X	X	X	X
Disable Job		X	X	X	X
Free Job		X	X	X	X
Assign Job		X	X	X	X
Unallocate Job		X	X	X	X
Allocate Job		X	X	X	X
Allocate Job to Plan Shift		X	X	X	X
Close Job		X	X	X	X
Despatch Job			X	X	X
Despatch Job to Shift			X	X	X
Acknowledge Job Despatch			X	X	X
Fix Job to Plan Depot		X	X	X	X
Fix Job to Plan Haul		X	X	X	X
Fix Job to Haul		X	X	X	X
Fix Job to Plan Resource		X	X	X	X
Fix Job to Resource		X	X	X	X
Delete Job					X
View Stop	X	X	X	X	X
Edit Stop				X	X
View Stop Time Window	X	X	X	X	X
Edit Stop Time WIndow				X	X
Despatch Stop			X	X	X
Stop Enroute			X	X	X
Stop Arrived			X	X	X
Stop Postponed			X	X	X
Start Stop			X	X	X
Complete Stop			X	X	X
Choose Shift for Stop		X	X	X	X
Relocate Stop			X	X	X

Client Operations at the default Access Levels (Continued) (Sheet 4 of 4)

Operations	VIEW_ ONLY	PLANNING	REAL_ TIME	DATA_ MOD	SYSTEM
Delete Stop					X
Send Text Message			X	X	X
Speed Time Windows					X
Cost Controls					X
Print Manifests	X	X	X	X	X

7.7.6 Client Connection/Disconnection status

Connect/Disconnect Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	A new Client has connected.	CONNECT
			A Client has disconnected.	DISCONNECT
IDENT	X	IDENT	Unique identifier for the Client.	
TYPE	X	TOKEN	Classifies the Client.	GUI, GATEWAY, SCHEDULER, NONE

SmAuto broadcasts a Client connection using the CONNECT packet. On dropping a Client connection SmAuto broadcasts the same using the DISCONNECT packet.

7.7.7 Specifying the Client View

View packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Tells SmAuto which partition of objects the connected Client wants to see.	VIEW
VIEW_ATTRIBUTES		IDENT	List of groups included in the view.	[ALL]
DATE_ATTRIBUTES		IDENT	List of dates included in the view. This field may contain both individual dates and date-ranges: 20060110, 20060112-20060114, 20060116-20060118	[ALL]
spare		STRING	May be used to extend VIEW packet without causing incompatibility with previous versions	
spare				
spare				
spare				

AllView Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Requests SmAuto for all the object views.	VIEW_ALL
VIEW_ATTRIBUTES	X	IDENT	All available views.	[NONE]

7.7.8 Default report behavior

SmAuto accepts and processes all packets a Client sends. However, different Clients have different data needs. On receipt of a “CLIENT_TYPE” packet, Client-Interface objects will be set to send only a subset of packets, depending on the type of the connected Client. As "Parameter" contains all information supplied by "Plan", these are mutually exclusive..

Default report behaviour

Aspect	Description
Message Report	Any message/command/request received by SmAuto is copied to the Client. **Note: Unlike other reports, messages are not accumulated when switched off.** **Note: Messages specifically destined for the Client will be received regardless.**
Object Report	An update to the problem definition. This involves sending everything that is known about the object(s).
Plan Report	An update to the generated schedule. These updates contain allocation data only. It is assumed that the recipient has already received the relevant problem data so that the schedule can be recreated.

Default report behaviour

Aspect	Description
Parameter Report	As “Plan Report”, but with additional parameters like ETA, ETD load etc.
Schedule Control	Parameters controlling the operation of the Scheduler.

Default Report behaviour for Client-Interfaces.:

ASPECT	None	Gui	Gateway	Scheduler
Message Report	X	X	X	X
Object Report		X		X
Plan Report				X
Parameter Report		X		
Schedule Control		X		X

7.7.8.1 Reporting Events

The REPORT_ON/OFF packets are used to switch the event-reports feature on or off.

Report ON/OFF packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Reports changes to the plan. Any accumulated changes are reported immediately.	REPORT_ON
			Disables reporting of changes. Changes are accumulated until the next REPORT_ON.	REPORT_OFF

SmAuto sends an AUTH_COMPLETE packet to report the termination of each authorization request.

Packet terminating Authorisation request

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Scheduler terminates each USER_AUTH_REQ with an AUTH_COMPLETE packet.	AUTH_COMPLETE

7.8 Oracle Real-Time Scheduler Control Packets

This section describes packets that control the Oracle Real-Time Scheduler scheduling process:.

Oracle Real-Time Scheduler Control Packets (Sheet 1 of 2)

Packet Type	Description
RESET	Resets the recipient to its initial state: no data, Scheduler off, real-time off, report off.
BATCH_ON	Advises the recipient that a stream of object updates are to follow.
BATCH_OFF	Advises the recipient that all object updates have been sent.
SCHED_ON	Enables Scheduler.
SPEED_TW_RESET	Clears all speed time window selections from the Client.
SCHED_OFF	Disables Scheduler.
BATCH_DONE	Advises the recipient that all object updates have been received and processed.
SR_BATCH_ON	Advises the recipient that a stream of Appointment Booking Request updates are to follow.
SR_BATCH_OFF	Advises the recipient that all Appointment Booking Request updates have been sent.
SR_BATCH_DONE	Advises the recipient that all Appointment Booking Request updates have been received and processed
CA_BATCH_ON	Advises the recipient that a stream of Conditional Assignment updates are to follow.
CA_BATCH_OFF	Advises the recipient that all Conditional Assignment updates have been sent.
CA_BATCH_DONE	Advises the recipient that all Conditional Assignment updates have been received and processed
COST_BATCH_ON	Advises the recipient that a stream of cost item updates are to follow.
COST_ITEM	Controls the objective function used by Scheduler.
COST_BATCH_OFF	Advises the recipient that a stream of cost item updates have been sent.
STATUS_BATCH_OFF	Advises the recipient that all cost status item updates have been sent.
STATUS_ITEM	Notifies the recipient of a status item.
STATUS_BATCH_ON	Advises the recipient that a stream of cost item update statuses are to follow. Provides total cost, cpu time and Scheduler status. Output from Scheduler only.
REALTIME	Specifies to what degree of Real-Time SmAuto operates.
SPEED_TW_PROFILE	Controls the average speeds by time-of-day.

Oracle Real-Time Scheduler Control Packets (Continued) (Sheet 2 of 2)

ACCEPT_CLIENTS	Enables Scheduler to accept Client connection. Scheduler receives this packet from Switch.
SEND_PLAN	Request SmAuto to send Data.
SYNC_PLAN	Request SmAuto to send Data Updates.
LOAD_DATA	Load view specifications.
SAVE_DATA	Save view specifications.

7.8.1 Scheduler Control

This section details the Packets that control the Scheduler operation.

Scheduler Control packets

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Resets Scheduler to its initial state.	RESET
			Delays all global consistency checks (plan validity checks) until BATCH_OFF is received. Also switches off reporting and Scheduler for the duration.	BATCH_ON
			Performs integrity-checks on all data received since BATCH_ON and processes where necessary.	BATCH_OFF
			Notifies the sender of receipt of all data and of completion of processing on the data.	BATCH_DONE
			Activates Scheduler. Scheduler will continually improve on the current plan until switched off.	SCHED_ON
			Clears all existing speed time window selections from the plannerClient. This packet must be bracketed by BATCH_ON and BATCH_OFF.	SPEED_TW_RESET
			Switches Scheduler off.	SCHED_OFF
			Enables Scheduler to accept Client connection, when Scheduler is not started as a server (that is "serverMode=FALSE). Scheduler receives this packet from Switch.	ACCEPT_CLIENTS

On receipt of BATCH_OFF, SmAuto will process the batch data. In order to notify the sender that all data has been received, and all processing has been completed, SmAuto will return a BATCH_DONE:

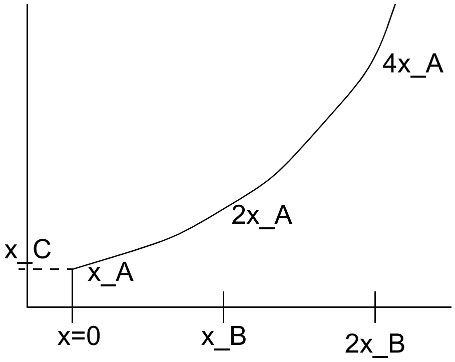
7.8.2 Cost Control

The costing aspect of Oracle Real-Time Scheduler schedules are controlled by the COST_BATCH_ON, COST_ITEM and COST_BATCH_OFF packets in order. The COST_ITEM packets set the various weights incorporated into the objective function of Scheduler. All fields are initialized to sensible values at start-up, allowing Scheduler to control a subset without needing to know the values of the remaining fields. An empty field leaves the weight unchanged.

The TRAVEL_DIST cost must be set for Scheduler from the command-line as there is no default value for this COST_ITEM. Unless changed by a COST_ITEM packet, the value set for TRAVEL_DIST cost may be used as a benchmark to gauge the effect of other parameters.

Many weights are controlled by three fields “x_A”, “x_B” and “x_C”. These weights represent an exponential function of x where “x_A” represents the initial slope for x>=0 (at x=0) and “x_B” represents the doubling rate. (the slope of the function doubles every “x_B”).:

Note: The number of slope-duplications can be set by the command-line option “maxExponent”. By default maxExponent is set to 8.



“x_C” represents a flat additional cost, creating an initial step. Within the cost-control packet descriptions, this function will be referred to as fABC(x).

7.8.2.1 Sentinel Cost Packets

COST_BATCH_ON and COST_BATCH_OFF are the sentinel packets within which COST_ITEM packets are sent to SmAuto.

Packet indicating the Start of Cost Items to come

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Specifies the starting point of COST_ITEM packets to be updated. Rejects the packet if no corresponding COST_BATCH_OFF packet is found.	COST_BATCH_ON

Packet conveying a single Cost Item

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Controls the objective function.	COST_ITEM
COST_NAME	X	TOKEN	Identifies the Cost parameter being sent.	See “Cost Item Parameters” on page 7 - 35.
COST_VALUES	X	STRING	Specifies the Cost parameter value. It is either a single value or a comma separated list depending on the cost type. See also section 7.8.2.2 “Cost Item Parameters” on page 7 - 35.	
TDC_VALUES		STRING	Specifies the time dependent cost factors. See also section 7.8.2.3 “Time Dependent Cost Factors” on page 7 - 41	A,B,C,D,E,F
spare		STRING	May be used to extend COST_ITEM without causing incompatibility with previous versions	
spare		STRING		
spare		STRING		

Packet terminating the Cost Items sent

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Indicates that no more COST_ITEM packets will be received	COST_BATCH_OFF

7.8.2.2 Cost Item Parameters

The Cost Item packets are sent every time a cost control profile is created or edited and applied. The HIP packets sent by Oracle Real-Time Scheduler also consist of Cost Item packets bounded by COST_BATCH_ON and COST_BATCH_OFF packets. Costs may be Time dependent costs (TDC) or non time dependent costs.

The following table contains the name, type, an indication of whether it is a TDC cost or not, a brief description and the acceptable value range for each cost item in Oracle Real-Time Scheduler.

Cost Item Parameters (Sheet 1 of 6)

COST_NAME	TYPE	TDC cost	Description	Range
RSRC	FLAT	No	Cost of using a Resource. Minimizes the total number of different Resources used.	[0], -MAXFLOAT, MAXFLOAT
HAUL	FLAT	No	The cost of using a Haul.	[0], -MAXFLOAT, MAXFLOAT
SHIFT	FLAT	Yes	Cost of using a Shift. Note: Applies a cost to the first “used” Shift for a Resource, and all subsequent Shifts up the last “used” Shift, inclusive.	[0], >=0
RUN	FLAT	No	Cost of activating a run.	[0], >=0
OVERTIME	EXPONENTIAL	Yes	Cost as a function of the number of seconds of over-time for a driver.	[0,MAXFLOAT,0], >=0, >=1, >=0
STOPLATE	EXPONENTIAL	Yes	Cost as a function of the number of seconds late at Stop.	[0,MAXFLOAT,0], >=0, >=1, >=0
WINDOW	FLAT	Yes	Cost of using a Time Window.	[0], >=0
STOP_SLA_WINDOW	FLAT	Yes	Cost of doing Job outside Stop SLA Window.	[0], >=0
DEPOT_SLA_WINDOW	FLAT	Yes	Cost of doing Job outside Depot SLA Window.	[0], >=0
DEPOTLATE	EXPONENTIAL	Yes	Cost as a function of the number of seconds late at Stop. In order to maintain balance with STOP_LATE, the resulting cost is multiplied with the number of Jobs transferred here.	[0,MAXFLOAT,0], >=0, >=1, >=0
LIFESPAN	EXPONENTIAL	Yes	Cost as a function of the number of seconds a product is on board longer than its life-span. LIFESPAN and MAX_RUN_LENGTH costs are mutually exclusive on actual application.	[0,MAXFLOAT,0], >=0, >=1, >=0

Cost Item Parameters (Continued) (Sheet 2 of 6)

COST_NAME	TYPE	TDC cost	Description	Range
VOLUME	EXPONENTIAL	Yes	Cost as a function of the **relative** number of units over max-volume. Calculated and added for each individual link.	[0,MAXFLOAT,0], >=0,>= 0.001,>= 0
WEIGHT	EXPONENTIAL	Yes	Cost as a function of the **relative** number of units over max-weight. Calculated and added for each individual link.	[0,MAXFLOAT,0], >=0,>= 0.001,>= 0
ZONE	FLAT	Yes	Flat cost for each Stop outside the Shift's zone.	[0], >=0
JOB_ATTR	FLAT	Yes	Cost of a Job-Job attribute preference mismatch.	[0], >=0
RSRC_ATTR	FLAT	Yes	Cost of a Resource-Job attribute preference mismatch.	[0], >=0
TRAVELDIST	FLAT	Yes	Cost of travel in units/m.	[NULL], >=0.001
TRAVELTIME	FLAT	Yes	Cost of time in seconds the driver spends "working" or taking break. Time spent at depot, logon, logoff (unless on break) is excluded. Any delays at Depot (loading, waiting) is not costed. Time spent on Breaks at Depot are costed TravelTime is costed on a "run" basis, i.e. Cost is applied to the time between departure from LogStops and Depots to the arrival at the next LogStop or Depot. When there is a break between the previous stop and the Depot/ LogStop, any waiting-time for the break is not costed, as it can be spent at depot/logoff. The latter is to prevent costing of time spent "waiting" for a break at depot. **Note:** The time spent **"on Break"** refers to Breaks actually scheduled or taken, not the total amount of Break-time available	[0], >=0
IDLETIME	FLAT	Yes	Cost against driver being idle in units/ second.	[0], >=0
LONGHAUL	FLAT	Yes	Standard cost of a Shift, which is part of a LONGHAUL, not terminating at the normal log-off position	[0], >=0
SITE	FLAT	Yes	Promotes Stops at the same site to be visited sequentially by a single Resource.	[0], >=0

Cost Item Parameters (Continued) (Sheet 3 of 6)

COST_NAME	TYPE	TDC cost	Description	Range
PKUP_SERVICE	EXPONENTIAL	Yes	Applies to PKUP, DIST Stops and PTP-Stops where load is picked up. Positive values bias towards early Collection, negative towards late Collection.	[0,MAXFLOAT,0], -MAXFLOAT, >=1, >=0
DROP_SERVICE	EXPONENTIAL	Yes	Applies to DROP, COLL Stops and PTP-Stops where load is dropped off. Biases towards early Distribution.	[0,MAXFLOAT,0], >=0, >=1, >=0

RUN_SEPARATION	EXPONENTIAL	Yes	Applies a cost to loading goods during a Distribution-run.
			cost = fABC(l/c) * r;
			l = load added at Stop.
			c = capacity available on Resource
			r = remaining Stops within the run.
				[0,MAXFLOAT,0], >=0,>= 0.001,>= 0

LOADLEVEL_SLOPE	FLAT	No	Cost as a function of the relative mismatch between load-levelling value and target.	[0], >=0
LOADLEVEL_MAX	FLAT	No	loadLevelCost when job-value == 0. By default is 5 * loadLevelCost.	[0], >=0
ALLOCATION	FLAT	No	The cost against each Stop that is not assigned, i.e. on the “free”-list.	[0], >=0
SITE_SEPARATION	FLAT	No	The cost against violating a site separation interval completely	[0], >=0
DEPOT_CAP_MAX	FLAT	No	The cost as function of the number of units over or under the target capacity of the Depot.	[0], >=0
DEPOT_CAP_ZERO	FLAT	No		[0], >=0
DEPOT_CAP_B1	FLAT	No		[1], 0.5, 20
DEPOT_CAP_B2	FLAT	No		[1], 0.5, 20
DEPWINCAP_VOLUME	EXPONENTIAL	No	Exponential cost for exceeding limit	[0,MAXFLOAT,0], >=0,>= 0.001,>= 0
DEPWINCAP_WEIGHT	EXPONENTIAL	No	Exponential cost for exceeding limit	[0,MAXFLOAT,0], >=0,>= 0.001,>= 0
PRODUCT_DEPOT	FLAT	No	The cost of using product-Depot capacity	[0], >=0
EMPTY_TRAVEL	EXPONENTIAL	Yes	Is the exponential cost for empty travel. The Range value represents the min values for A, B, C respectively	[0,MAXFLOAT,0], >=0, >=1, >=0

Cost Item Parameters (Continued) (Sheet 4 of 6)

COST_NAME	TYPE	TDC cost	Description	Range
ROUND_ZONE	FLAT	Yes	Flat cost for each run with no Stops within the Shift's zone. This cost is calculated in conjunction with Zone Cost. See “ZONE” on page 7 - 36	[0], >=0
APEX	FLAT	Yes	Cost for not finishing a run at the Apex. Encourages the furthest Stop from the Depot to be the last Stop in the run. Apex cost is applied only for runs where the distance is greater than that of “Apex Run Distance” set from Oracle Real-Time Scheduler's Launcpad. Apex cost is not applied for closer runs (default = 10km).	[0], >=0
ROUND_LENGTH	FLAT	Yes	Cost of time spend from first to last Job-Stop of a run. Promotes geographical clustering of Stops in a run.	[0], >=0
ROUND_LENGTH_VAR	FLAT	No	Cost of variation in total runLength across a Shift	[0], >=0
ROUND	FLAT	Yes	Cost for the geographical size of run. Promotes geographical clustering of Stops in a run.	[0], >=0
ROUND_VAR	FLAT	No	Cost for variance in run area across a Shift	[0], >=0
ROUND_OL	FLAT	No	Cost for overlap in runArea's	[0], >=0
ROUND_END_OL	FLAT	No	Cost against runs finishing in the same Run End Area.	[0], >=0
WAIT_TIME	FLAT	Yes	Cost for being driver idle	[0], >=0
WAIT_TIME_PAST	FLAT	Yes	Cost for driver being idle in the past.	[0], >=0
BREAK_LATE	EXPONENTIAL	Yes	Cost for having taken a break late.	[0,MAXFLOAT,0], >=0, >=1, >=0

Cost Item Parameters (Continued) (Sheet 5 of 6)

COST_NAME	TYPE	TDC cost	Description	Range

RESERVE_CAPACITY	EXPONENTIAL	Yes	Exponential cost for exceeding shift's reserve capacity.
			reserveCapacityCost = RESERVE_CAPACITY_A * (TOTALCONSUMEDCAPACITY - TOTALAVAILABLECAPACITY).
			TOTALDELAY = SUM(JOBDELAY, SITEDELAY of all stops with matching CAPACITY_ID as Shift)
			TOTALAVAILABLECAPACITY = MAXSHIFT - (Shift's RESERVE_CAPACITY * MAXSHIFT) + TOTALDELAY
			TOTALCONSUMEDCAPACITY = SHIFT_LENGTH - Shift's Idle_Time.
				[0,MAXFLOAT,0], >=0,>= 0.001,>= 0

SHIFT_PROMOTION	EXPONENTIAL	Yes	Shift-promotion costs introduce a cost against unnecessarily delaying a Job. Normally this cost would rise quickly for each of the first few days, then level off. This has the effect of heavily promoting the first few days in order to fill up partly filled routes, and if not successful, concentrate on a good fit with routes further downstream that are still "under construction". This is a time-dependent cost, and the value of this cost in each shift will be adjusted by the TDC function. The shift-promotion cost is the product of the stop's multiplier and shift's time-based promotion cost. So changing the order of stops in the shift will not change the shift-promotion cost. **Note:** In view of Longhaul shifts, Shift-promotion cost is defined as a dynamic cost. This way the stop can be moved to different shifts, thereby generating different shift promotion costs.	[0], >=0
JOBS_LIMIT	EXPONENTIAL	Yes	Sets the cost of exceeding the shift's maximum Job limit. **For example:** This cost can be used to restrict the maximum number of Jobs that are assigned to a field technician in a single shift.	[0,MAXFLOAT,0], >=0,>= 0.001,>= 0

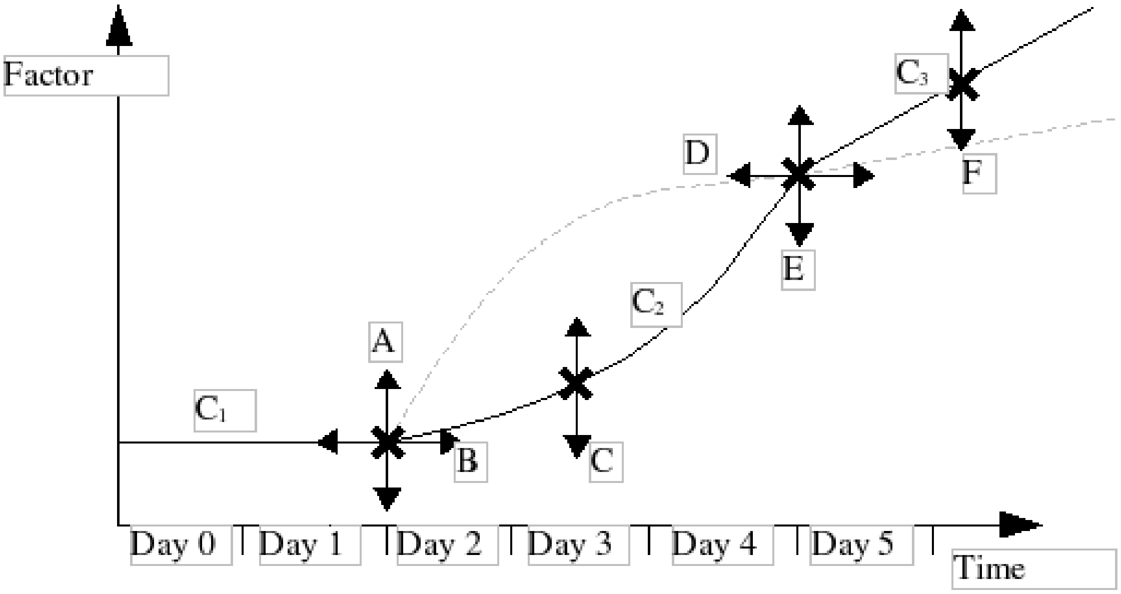
Cost Item Parameters (Continued) (Sheet 6 of 6)

COST_NAME	TYPE	TDC cost	Description	Range
DYNAMIC_WORKING_ AREA	EXPON ENTIAL	Yes	The "Dynamic Work Areas" functionality is used to discourage planner from using a Driver who is not closest for a Job. ("closest" is determined by the "travel time" between "logon" position and the Job location). When a Job is entered into planner, the "closest" driver/resource is established, and thereafter the dynamic working area "cost" is applied if the Job is assigned to a Driver who "lives" further away. The cost is based on the difference in "travel time". This cost is applied when the vehicle for a Stop is based further from the Stop than from another equally suitable vehicle. The exponential cost parameters A represents strength and B represents interval (in secs) at which this cost value is duplicated. The lower the B-value is, the lower the overlapping in working zones for vehicles that is permitted by Planner.	[0,MAXFLOAT,0], >=0, >=1, >=0

Note: MAXFLOAT = maximum real number in the system. (maxA = maxB = maxC = MAXFLOAT for all exponential costs) Based on cost type each COST_ITEM has either one or three variables. Each of these variables in turn has its own range. Flat costs have a single variable, while exponential costs have three variables.

Warning: RSRC, HAUL and PKUP_SERVICE are the only three cost items for which Scheduler will accept negative values. All other cost items must be greater than zero.

7.8.2.3 Time Dependent Cost Factors



The configuration of the Time Dependent Cost Factors is quite complex. There are three distinct curves which can be defined individually. The initial curve (C1) is always flat. The second curve (C2) is either convex or concave. The final curve (C3) is always a straight line into infinity. In order to define this shape 6 independent variables are required. These variables are specified as time dependent cost factors in the TDC_VALUES field of the COST_ITEM packet.

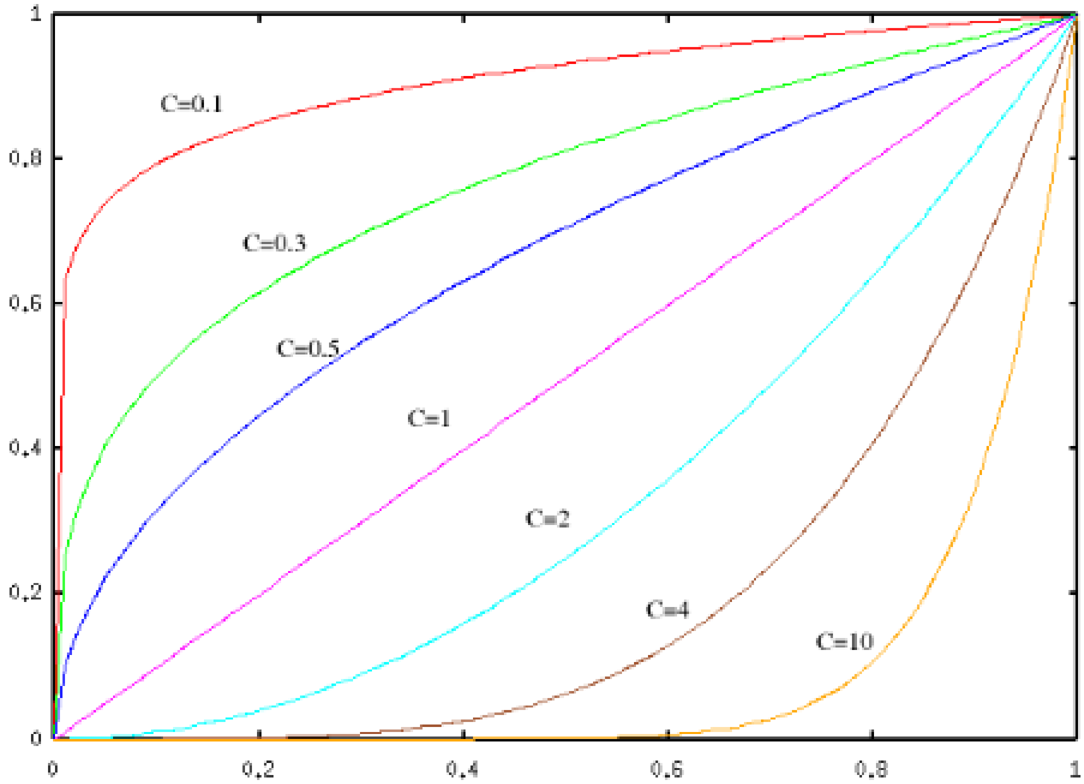
7.8.2.3.1 Time Dependent Cost Factor Definition

1 **A** Initial factor, flat from Day 0 to B. Range >=0

C_1 [x:0..B) = A

2 **B** Number of Hours after which function changes from flat to curved. Range >=0

3 **C** Value which defines the shape of the curve. Range 0.1<>10



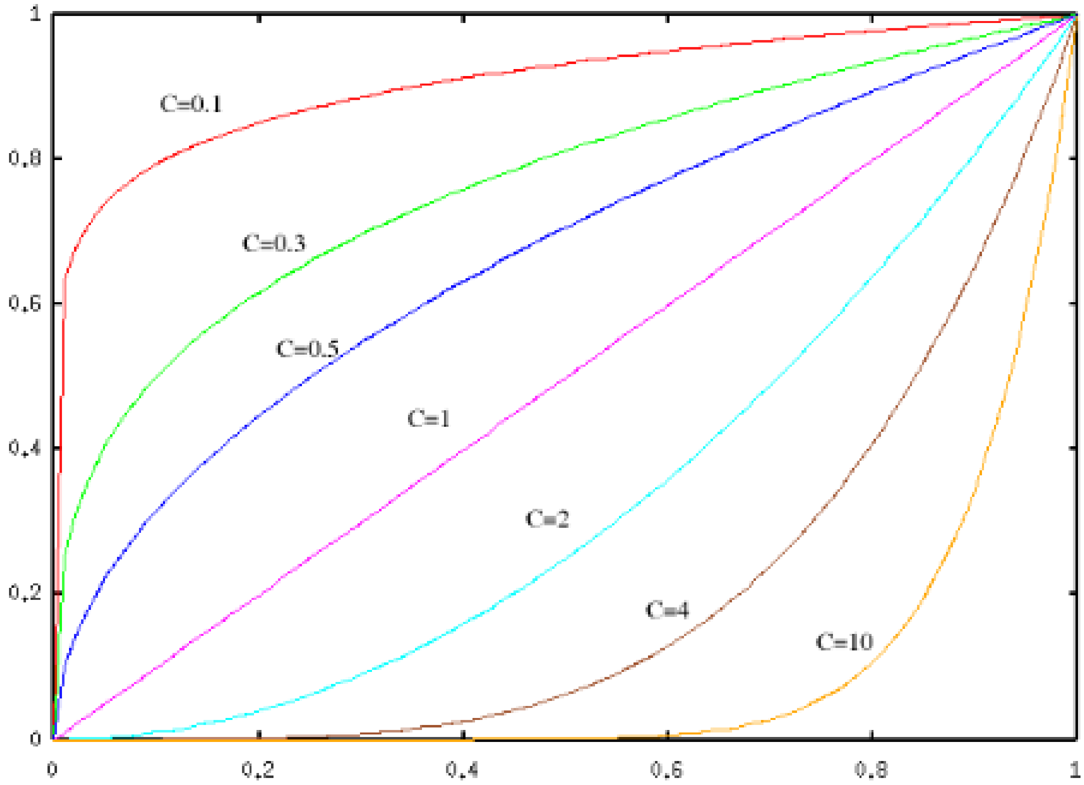
C_2 [x:B..D) = $((x - B) / (D - B))^C * (E - A) + A$

4 **D** Number of hours after which function changes from curved to linear. Range >B

5 **E** Factor at the point at which the function changes from curved to linear. Range >=0

6 **F** Slope of linear function

C3 [x:D..) = $(x - D) * F + E$


C=0.1
C=0.3
C=0.5
C=1
C=2
C=4
C=10
0
0.2
0.4
0.6
0.8
1
0
0.2
0.4
0.6
0.8
1

7.8.3 The Schedule’s Cost Status

Once the costs are applied to the plan, Scheduler advises all Clients of any cost changes through a series of STATUS_ITEM packets at regular intervals while in SCHED mode and after every Scheduler state change. Each cost is represented as a STATUS_ITEM packet and sent as a batch within a STATUS_BATCH_ON and STATUS_BATCH_OFF packet

Packet indicating the Start of "Schedule Cost Items Status" to come

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Scheduler notifies the Client that a stream of COST_ITEM update statuses are to follow. Provides total cost, cpu time and Scheduler status.	STATUS_BATCH_ON
STATUS	X	TOKEN	Current Scheduler-status.	See “Scheduler Status” on page 48. for the different statuses Scheduler can exist in.
CPU_TIME	X	STRING	CPU-time used since last RESET-message. Needed for “nice” graph on Oracle Real-Time Scheduler.	“HH:MM:SS.CC”, where CC is 100ths of seconds
REMAIN	X	DOUBLE	Indication for the time remaining. Scheduler switches to IDLE on reaching 0.	
TOTAL_COST	X	DOUBLE	Total cost of current solution.	>=0
DATE		DATE	Current date on Scheduler	
TIME		HHMMSS	Current time of day on Scheduler	"000000-235959" If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)), that is: ((local time - UTC offset) to (local time + UTC offset))
Spare		STRING	May be used to extend STATUS_BATCH_ON without causing incompatibility with previous versions	
Spare		STRING		
Spare		STRING		

Tokens indicating the Schedule's cost item status

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Scheduler notifies the Client of a received and processed COST_ITEM status.	STATUS_ITEM
STATUS_NAME	X	TOKEN	Displays the status name of the corresponding COST_ITEM after Scheduler has received and processed it.	See "Status Item Parameters" on page 7 - 45.
STATUS_VALUE	X	TOKEN	Displays the status value (cost) of the corresponding COST_ITEM after Scheduler has received and processed it.	

Packet terminating the Cost Items Status tokens

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Confirms that all STATUS_ITEM updates have been received and processed.	STATUS_BATCH _OFF

7.8.3.1 Status Item Parameters

The Status Item Parameters are listed below.

Status Item Parameters (Sheet 1 of 4)

STATUS_NAME	Format	Description	Controlling Cost Item(s)
VEHICLE_COST	DOUBLE	Cost of using a Resource. Minimizes the total number of different Resources used.	RSRC
HAUL_COST	DOUBLE	The cost of using a Haul.	HAUL
SHIFT_COST	DOUBLE	Cost of using a Shift.	SHIFT
RUN_COST	DOUBLE	Cost of activating a run.	RUN
OFFLATE_COST	DOUBLE	Cost as a function of the number of seconds of over-time for a driver.	OVERTIME
HAUL_EXTEND_COST	DOUBLE	Cost against the Haul extending over a specified amount of time (maxHaul in Shift).	OVERTIME
STOPLATE_COST	DOUBLE	Cost as a function of the number of seconds late at Stop.	STOPLATE
WINDOW_COST	DOUBLE	Cost of doing Job outside Stop or Depot Access Window.	WINDOW

Status Item Parameters (Continued) (Sheet 2 of 4)

STATUS_NAME	Format	Description	Controlling Cost Item(s)
STOP_SLA_WINDOW_COST	DOUBLE	Cost of doing Job outside Stop SLA Window.	STOP_SLA_WINDOW
DEPOT_SLA_WINDOW_COST	DOUBLE	Cost of doing Job outside Depot SLA Window.	DEPOT_SLA_WINDOW
DEPOTLATE_COST	DOUBLE	Cost as a function of the number of seconds late at Stop. In order to maintain balance with STOP_LATE, the resulting cost is multiplied with the number of Jobs transferred here.	DEPOTLATE
LIFESPAN_COST	DOUBLE	Cost as a function of the number of seconds a product is on board longer than its life-span.	LIFESPAN
LOAD_COST	DOUBLE	Cost as a function of the **relative** number of units over max (weight + volume). Calculated and added for each individual link.	WEIGHT, VOLUME
ZONE_COST	DOUBLE	Cost of a Resource-Job zone preference mismatch.	ZONE
JOB_ATTR_COST	DOUBLE	Cost of a Job-Job attribute preference mismatch.	JOB_ATTR
RSRC_ATTR_COST	DOUBLE	Cost of a Resource-Job attribute preference mismatch.	RSRC_ATTR
DIST_COST	DOUBLE	Cost of travel in units/m.	TRAVELDIST
TRAVELTIME_COST	DOUBLE	Cost of time in seconds the driver spends "working" or taking break. Time spent at depot, logon, logoff (unless on break) is not costed.	TRAVELTIME
IDLETIME_COST			IDLETIME
LONGHAUL_COST	DOUBLE	Standard cost of a Shift, which is part of a LONGHAUL, not terminating at the normal log-off position	LONGHAUL
SITE_COST	DOUBLE	Promotes Stops at the same site to be visited sequentially by a single Resource.	SITE
PKUP_SERVICE_COST	DOUBLE	Applies to PKUP, DIST Stops and PTP-Stops where load is picked up. Positive values bias towards early Collection.	PKUP_SERVICE
DROP_SERVICE_COST	DOUBLE	Applies to DROP, COLL Stops and PTP-Stops where load is dropped off. Biases towards early Distribution.	DROP_SERVICE

Status Item Parameters (Continued) (Sheet 3 of 4)

STATUS_NAME	Format	Description	Controlling Cost Item(s)

RUN_SEPARATION_COST	DOUBLE	Applies a cost to loading goods during a Distribution-run.
		cost = fABC(l/c) * r;
		l = load added at Stop.
		c = capacity available on Resource
		r = remaining Stops within the run.
			RUN_SEPARATION

LOAD_LEVEL_COST	DOUBLE	Cost as a function of the relative mismatch between load-levelling value and target.	LOADLEVEL_SLOPE, LOADLEVEL_MAX
ALLOCATION_COST	DOUBLE	The cost against each Stop that is not assigned, i.e. on the “free”-list.	ALLOCATION
SITE_SEPARATION_COST	DOUBLE	The cost against violating a site separation interval completely	SITE_SEPARATION
DEPOT_CAPACITY_COST	DOUBLE	The cost as function of the number of units over or under the target capacity of the Depot.	DEPOT_CAP_ZERO, DEPOT_CAP_MAX, DEPOT_CAP_B1, DEPOT_CAP_B2
PRODUCT_DEPOT_COST	DOUBLE	The cost of using product-Depot capacity	PRODUCT_DEPOT
EMPTY_TRAVEL_COST	DOUBLE	Is the exponential cost for empty travel. The Range value represents the min values for A, B, C respectively	EMPTY_TRAVEL
APEX_COST	DOUBLE	Cost for not finishing a run at the Apex. Encourages the furthest Stop from the Depot to be the last Stop in the run. Apex cost is applied only for runs where the distance is greater than that of “Apex Run Distance” set from Oracle Real-Time Scheduler’s Launcpad. Apex cost is not applied for closer runs (default = 10km).	APEX
ROUND_LEN_COST	DOUBLE	Cost of time spend from first to last Job-Stop of a run. Promotes geographical clustering of Stops in a run.	ROUND_LENGTH
ROUND_LEN_VAR_COST	DOUBLE	Cost of variation in total runLength across a Shift	ROUND_LENGTH_VAR
ROUND_COST	DOUBLE	Cost for the geographical size of run. Promotes geographical clustering of Stops in a run.	ROUND
ROUND_VAR_COST	DOUBLE	Cost for variance in run area across a Shift	ROUND_VAR

Status Item Parameters (Continued) (Sheet 4 of 4)

STATUS_NAME	Format	Description	Controlling Cost Item(s)
ROUND_OL_COST	DOUBLE	Cost for overlap in runArea’s	ROUND_OL
ROUND_END_OL_COST	DOUBLE	Cost against runs finishing in the same Run End Area.	ROUND_END_OL
WAIT_COST	DOUBLE	Cost for being driver idle	WAIT
~~WAIT_TIME_PAST_COST~~	~~DOUBLE~~	~~Cost for driver being idle in the past.~~	~~WAIT_TIME_PAST~~
BREAK_EARLY_COST	DOUBLE	Cost for having taken a break early.	BREAK_EARLY
BREAK_LATE_COST	DOUBLE	Cost for having taken a break late.	BREAK_LATE
RESERVE_CAPACITY_COST	DOUBLE		RESERVE_CAPACITY
MANYSHIFT_COST	DOUBLE	Cost against carrying leftover goods in Resource after completing Shift. Default cost of (10^90).	These are exceptional costs that must be zero under normal operation. Very large costs can result from the presence of these costs in the plan. Hence these costs are fixed in the Oracle Real-Time Scheduler Scheduler process and the user is not allowed to change them.
COMPAT_COST	DOUBLE	Cost against having Stop/Shift incompatibilities in the plan. Default cost of (10^90).	
MAP_CONNECT	DOUBLE	Cost against using disconnected map Default cost of (10^90).	

The Scheduler exists in one of the following States at any given time:

Scheduler Status

Token	Value
OFF	Not active.
IDLE	Activated, but has not yet commenced a task.
INIT	Calculating an initial plan.
SCHED	Improving the current plan.
TERM	Completed its task, but changes to the “problem” data may cause it to re-commence.

7.8.4 Real-Time

The REALTIME packet synchronises the Scheduler's "Real-Time" status with that of all connected clients. A Client sends a RELATIME packet with modified status to the Scheduler. The Scheduler changes status accordingly and notifies all connected Clients of the status change. If the Real-Time status set in the Client is different to that of the Scheduler, a warning will be generated to notify the user that the Real-Time state has changed.

Packet Specifying the Scheduler's Real-Time Mode

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Customizes SmAuto's behavior with regard to Real-Time.	REALTIME
TYPE		TOKEN	Real-Time mode. See "Real-Time modes" on page 49.	[OFF], ~~STANDARD,~~ AUTO_POSITION, AUTO_COMPLETE

Real-Time modes

Mode	Description
OFF	In this state Scheduler does not consider the current time, does not move Resource's and can schedule Stops to be started in the past. Except for recommendations made by itself, SmAuto will not modify any of the data supplied unless requested to do so.
STANDARD	New or updated Jobs are re-assigned immediately.
AUTO_POSITION	In this state Scheduler automatically advances STANDARD + Resources by way of dead-reckoning till the next uncompleted Stop and only schedules Stops to be started in the future.
AUTO_COMPLETE	This state is similar to Auto Position with the added functionality that Scheduler assumes Resources are arriving, starting and completing work as planned. AUTO_POSITION + uncompleted Stops are automatically completed as planned.

Up until Oracle Real-Time Scheduler 10.0 various items of functionality within Oracle Real-Time Scheduler such as APE (Allocation Priority Escalation) and Reserve Capacity were using WallTime (The current time as indicated by the clock on the wall) to influence behaviour when in real-time mode.

The only way to recreate past behaviour for these items of functionality were to use Vdate/Vtime/Vwarp to simulate the passage of time. Often, because of the difficulties in synchronizing a data-stream with this "fake" time, this was not a suitable approach for debugging and modeling.

In Oracle Real-Time Scheduler 4.2 a referenceTime has been introduced to be used by these processes instead of WallTime. While in Real-Time mode, Planner updates referenceTime from WallTime, writing a referenceTime packet together with any other data such as SYNC_PLAN and AutoSave. When in BatchMode Planner reads the referenceTime packets associated with any data, allowing APE etc. to be properly configured.

Reference Time packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	In Real-Time mode this time is used as the current time to model the passage of time.	REFERENCE_TIME
START_DATE		DATE	Is the date of the referenceTime	
START_TIME		HHMMSS	Is the time of day of the referenceTime	"000000-235959" If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)). That is: ((local time - UTC offset) to (local time + UTC offset))

7.8.5 Speed Time Window Profiles

Multiple speed time window profiles enable you to record the variability in road speeds during different times on a day by day basis. Speed time window profiles can be maintained as "default" (DEFAULT) or belonging to a "day" of the week (MON, TUE, WED, THU, FRI, SAT, SUN) or as per the requirements of a specific "date" (yyyymmdd).

Speed TW Profile Packet (Sheet 1 of 2)

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Consists of an array of speed time-windows such that during each of these time window the average speed differs from the map.	SPEED_TW_PROFILE
SPEED_PROFILE_TYPE	X	IDENT	It is one of DATE, DAY or DEFAULT. DATE must be a valid date. DAY must be the first three letters of the day of week in uppercase letters.	DEFAULT
SPEED_PROFILE_NAME	X	IDENT	The speed profile name that will be available for the CLient	

Speed TW Profile Packet (Continued) (Sheet 2 of 2)

Field Name	M	Format	Description	Range
SPEED_PROFILE_GRANULARITY		HHMM	Period of time when relative speed will remain constant. This value is currently hard-coded as 15 minutes in the system.	15
REL_SPEEDS		IDENT	Is an array of relative speeds one each for a granularity such that it covers a whole day period. As granularity is currently fixed at 15 minutes the number of relative speeds in this array is also fixed at ((24*60)/15) = 96. Relative speeds must be in the range >0 and <=2 and is specified as a comma separated list of 96 items. For example: 0.5,0.2,,0.7,1.6,,,,,,,,......,96thvalue Relative speed of 1 will automatically be substituted wherever there is no value before a comma.	Relative speeds must be a value in the range >0 and <=2
spare		STRING	May be used to extend SPEED_TW_PROFILE without causing incompatibility with previous versions	
spare		STRING		
spare		STRING		

7.8.5.1 Error Control in speed time windows

SPEED_PROFILE_GRANULARITY is hard-coded at 15 minutes and is used as the interval at which speed time windows will be generated automatically for a 24 hour day. Consistency in speed time window profiles if further enhanced by not allowing overlapping speed time windows within profiles of the same type. If multiple speed time window profiles of the same type are loaded into Oracle Real-Time Scheduler, then these will be rejected and an error message will be issued.

It is not possible to store speed time windows exceeding 24 hours duration into a single profile. If the same speed should be applied after midnight on a specific date or day of the week, a separate profile for this day must be sent to Oracle Real-Time Scheduler.

It is not necessary to specify the speed for every interval in the 24 hour period. Relative speeds that have not been specified during any time intervals will be defaulted to the next lower priority profile as follows:

1. Relative Speed of corresponding time interval from the matching date profile
2. Relative Speed of corresponding time interval from the matching day of week profile
3. Relative Speed of corresponding time interval from the DEFAULT profile which will be 1 unless specified differently.

7.8.5.2 Backward Compatibility

To maintain backward compatibility with pre Oracle Real-Time Scheduler4.2 interface versions, only the speed time windows associated with the DEFAULT speed time windows profile will be saved for all Oracle Real-Time Scheduler interface versions lower than 4.2.

7.8.6 Data transfer

These packets control loading and saving of data to Host or file. In general the Allocator will initiate a transfer by sending a LOAD_DATA packet containing a specification of what to load through GateWay. (The format of the specification is as yet not determined).

GateWay decodes the specification and requests the data from SmAuto by way of a SEND_PLAN or SYNC_PLAN packet. The Allocator can then select a valid specification and again send LOAD_DATA.

Data Transfer packets

Packet Type	Description
LOAD_DATA	A request for GateWay to load the specified data into SmAuto.
SAVE_DATA	A request for GateWay to prepare for receipt of specified data.
SEND_PLAN	A request for Scheduler to send specified allocation information. SmAuto will send all available objects as requested, within the Report_ON/OFF packets
SYNC_PLAN	A request for Scheduler to send specified allocation information. SmAuto will send only those objects necessary to synchronize the Client with the current plan.

The following packets support the Load/Save operations through GateWay

Load/Save view specifications

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Requests GateWay to load the specified (or all) data.	LOAD_DATA
			Requests GateWay to save the specified (or all) data.	SAVE_DATA
ORIGIN		STRING	Identifies the Client that originally sent the message. Tells GateWay where to send error-messages.	
VIEW		IDENT	Specifies the required data.	

The SEND_PLAN packet is used by GateWay to prompt SmAuto to send the data GateWay wants to save. On receipt, SmAuto sends ALL requested data (bracketed by BATCH_ON/OFF)

Request for Scheduler to send data

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Requests SmAuto to send the specified plan-data (PLAN_STOP....). To prevent modifications to the plan while transmitting, the send must be completed as an **atomic** operation.	SEND_PLAN
VIEW_ATTRIBUTES		IDENT	Attributes of the objects to be sent.	[]
TIME_VIEW		DATE	Date of the view. If not specified, all data will be selected.	[]

The SYNC_PLAN packet is used by a Client to prompt SmAuto to send all the data necessary to synchronize the Client with SmAuto. The data is bracketed by BATCH_ON/OFF packets.

Request for Scheduler to synchronize Client

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Requests SmAuto to send the specified plan-data (PARAM_STOP....). To prevent modifications to the plan while transmitting, the send must be completed as an **atomic** operation.	SYNC_PLAN

7.9 OBJECT PACKETS

This section describes the Object packets used to build a model of the scheduling problem to be solved within SmAuto.

Object data consists of Resources, Resource time-windows, Stops, Stop time-windows, Depots and Depot time-windows. On top of begin and end of the window, time-windows may contain additional parameters applicable to the window.

Object packets

Packet Type	Description
DEPOT	Defines the Depot.
DEPOT_TW	Defines an access time-window for the Depot.
DEPOT_SLA_TW	Defines a service level agreement time-window for the Depot
RSRC	Defines/updates the Resource.
SHIFT	Defines the Resource Shift.
STOP	Defines the Stop.
STOP_TW	Defines an access time-window for the Stop.
STOP_SLA_TW	Defines a service level agreement time-window for the Stop.
STOP_AP	Defines the Allocation Priority configuration for the Stop.
BREAK	Defines the Break.
PRODUCT_DEPOT	Defines the product-Depot relationship.
SLOT	Adds/replaces the Slot.

7.9.1 Depot

The DEPOT packet describes a Depot by a unique ID, type, location and other attributes. The Scheduler will generate Stops (DEPOT_STOP) at a Depot as necessary, using the DEPOT record as a template.

Depot Packet (Sheet 1 of 2)

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Defines a Depot.	DEPOT
IDENT	X	IDENT	Uniquely identifies the Depot.	
TYPE	X	TOKEN	Defines the type of Depot. See DepotTypes table below.	DIST, COLL
AUTOCOMPL_MODE		TOKEN	When REALTIME is set to AUTO_COMPLETE, Stops at this Depot will be automatically “completed” as scheduled.	[YES],NO
LATITUDE	X	DOUBLE	Position of the Depot. Accurate positioning needs 6 significant digits after the decimal.	
LONGITUDE				
SITE_DELAY		CARDINAL	Fixed delay associated with transferring any number of Jobs at this Depot.	>=0; [0 secs]
RELSPEED		DOUBLE	Relative speed in the area. Used to calculate travel-time from the nearest known point on the map to the site.	>=0.01, <=100; [1.0]
VIEW_ATTRIBUTES		IDENT	Attributes determining which Client has this Depot in its “view”.	[ALL]
FOREIGN		TOKEN	Indicates whether this Depot is part of the schedule, or is a placeholder for a Depot excluded from the schedule. That is, “Foreign Depots” are not part of the schedule, but are displayed to the Allocator, who may assign Jobs to “Foreign Depots” in the same manner as if they were local Depots. **Note:** When a Job has been assigned to a “Foreign Depot”, it cannot be part of the schedule, and Oracle Real-Time Scheduler disables and puts it in the “free” list.	[NO],YES

Depot Packet (Continued) (Sheet 2 of 2)

Field Name	M	Format	Description	Range
spare		STRING	May be used to extend DEPOT without causing incompatibility with previous versions	
spare		STRING		
spare		STRING		
spare		STRING		
spare		STRING		
spare		STRING		
Label		STRING	This the DEPOT label	Not used by Oracle Real-Time Scheduler. For display only.
Location		STRING	Can be the suburb, town, locality	
Street_No		STRING	This is part of the Depot address	
Street_Name		STRING	This is part of the Depot address	
Postcode		STRING	This is part of the Depot address	
Location_Suburb		STRING	e.g. St. Andrews, FIFE	
State		STRING	e.g. Victoria	
Country		STRING	e.g. Canada	
Contact_Phone		STRING	A telephone number for the Depot	
Contact_Name		STRING		
Special_Conditions		STRING	i.e. “West entrance blocked”	
Special_Text		STRING	i.e. “Good Morning”	
COMMENT		STRING	Comments, e.g. “Closed for lunch”	

Oracle Real-Time Scheduler recognises the following Depot Types

Token	Description
DIST	Represents a Depot from where goods are picked up for Distribution to individual customer-sites.
COLL	Represents a Depot at which goods collected from individual customer-sites are delivered.

7.9.2 Depot Access TW

Time-windows for Depots are classified as Access Time Windows (defined by DEPOT_TW packets) and Service Time Windows (defined by DEPOT_SLA_TW packets). It is mandatory for a valid Depot to at least have an AccessTW. There is no access to the Depot outside of the AccessTW.

MAXWEIGHT and MAXVOLUME fields in the Depot_TW packet aid in determining the Depot capacity during a specific time-window.

Depot Access time-window Packet (Sheet 1 of 2)

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Defines a Depot time-window.	DEPOT_TW
IDENT	X	IDENT	Uniquely identifies the window.	
DEPOT	X	IDENT	The Depot to which the window is assigned.	
STATUS		TOKEN	Indicates whether the window may be used or not.	[ACTIVE] INACTIVE
START_DATE	X	DATE	Starting Date, on which the Depot Time Window can be accessed.	yyyymmdd
START_TIME		HHMM	Time of day at which the Depot Time Window will start	hhmm If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)), that is: ((local time - UTC offset) to (local time + UTC offset))
DURATION		HHMM	Period of time from the start of its Time Window, that the Depot will be open.	hhmm
WINDOW_COST		DOUBLE	Relative cost to visit within the window.	>= 0; [0.0]
LATE_COST		DOUBLE	Relative cost arriving late.	>=0.001,<=1e10; [1.0]
MAXWEIGHT		DOUBLE	Maximum total load that can be delivered to/collected from the Depot during this time-window.	[unlimited]
MAXVOLUME		DOUBLE		[unlimited]

Depot Access time-window Packet (Continued) (Sheet 2 of 2)

Field Name	M	Format	Description	Range
spare		STRING	May be used to extend Depot_TW without causing incompatibility with previous versions	
spare				
spare				
spare				
spare				
spare				

7.9.3 Depot Service TW

Depot Service Time Window specifies the time that has been agreed upon for servicing a Depot based Job. Also see section 7.9.2 “Depot Access TW” on page 7 - 57.

Oracle Real-Time Scheduler will preferably try to schedule the Job within the SLA Time Window or else incur a cost. The SLA_FLEXIBILITY field specifies the time period before and after the SLA Time Window outside which when the Job is done a fixed predefined cost is applied.

Depot Service time-window Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Defines a Depot time-window.	DEPOT_SLA_TW
IDENT	X	IDENT	Uniquely identifies the window.	
DEPOT	X	IDENT	The Depot to which the window is assigned.	
STATUS		TOKEN	Indicates whether the window may be used or not.	[ACTIVE] INACTIVE
START_DATE	X	DATE	Date on which the Depot SLA Time Window can be accessed.	yyyymmdd
START_TIME		HHMM	Time of day when the Depot’s SLA Time Window will begin	hhmm If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)), that is: ((local time - UTC offset) to (local time + UTC offset))
DURATION		HHMM	Period of time from the start of its SLA Time Window, that the Depot will be open.	hhmm
SLA_FLEXIBILITY		HHMM	Defines the period of time before and after SLA Time Window such that, arriving earlier/later than this time will compulsorily apply a fixed maximum cost.	hhmm
spare		STRING	May be used to extend Depot_TW without causing incompatibility with previous versions	
spare				
spare				
spare				
spare				
spare				

7.9.4 Product-Depot

The PRODUCT_DEPOT packet defines the products which are available at Depot

Product-Depot Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Depot-product relation	PRODUCT_DEPOT
PRODUCT_ID	X	IDENT	Product available at Depot.	
DEPOT_ID	X	IDENT	Depot this product is related to	
MAX_UNITS		DOUBLE	Maximum capacity (units) of product available at this Depot	>=0
TARGET		DOUBLE	Target Units of this product to be shipped/collected at this Depot.	>=0;[0]
PRODUCT_COST		DOUBLE	relative product cost	>=0;
TARGET-PRIORITY		DOUBLE	cost of meeting target value	>=0;
spare			May be used to extend PRODUCT_DEPOT without causing incompatibility with previous versions	
spare				
spare				

7.9.5 Stop

The STOP-packet defines properties of a Stop but not plan-related info like “allocated Shift”.

Stop Packet (Sheet 1 of 8)

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Defines a Stop.	STOP
IDENT	X	IDENT	Uniquely identifies the Stop.	
JOB		IDENT	Identifies the JOB the Stop belongs to, if any.	[IDENT]
JOB_STOP_ORDER		CARDINAL	Individual Stops that are part of the same JOB will be visited in order of JOB_STOP_ORDER.	[0]
DESPATCH_MODE		TOKEN	Defines how the Job will be despatched.	[MANUAL] AUTO
AUTOCOMPL_MODE		TOKEN	When REALTIME is set to AUTO_COMPLETE, this Stop will be automatically “completed” as scheduled.	[YES],NO

Stop Packet (Continued) (Sheet 2 of 8)

Field Name	M	Format	Description	Range
INITIAL_STATUS		TOKEN	The initial status of a new Job. Scheduler inserts Jobs with FREE status into the plan, while Jobs with INACTIVE status must be manipulated manually by the operator. See “Conditional Assignment of Job” on page 7 - 126.	[FREE],INACTIVE
JOB_CODE		IDENT	A code that may be used to identify the Job within an MDT.	[JOB]
STOP_CODE		IDENT	A code that may be used to identify the Stop within an MDT.	[IDENT]
TYPE	X	TOKEN	Defines the type of Stop. If the type is PKUP or DROP, Depot must be set to specify suitable Depots. Periods of Unavailability will be defined as POU stops.	PKUP, DROP, STOP, POU
DEPOT		IDENT	“Comma” separated list of Depots where goods may be collected from or dropped off at. **Note:** All Stops belonging to same Job must specify the same set of Depot.	[none]
LATITUDE	X	DOUBLE	Position of the Stop. Accurate positioning needs 6 significant digits after the decimal.	
LONGITUDE				
SITE		IDENT	Identifies the site so the SmAuto can correctly calculate total delay when doing several Jobs at the same site.	[IDENT]
SITE_DELAY		CARDINAL	Fixed delay associated with transferring any number of Jobs at this site.	>=0; [0 secs]
JOB_DELAY		CARDINAL	Delay associated with transferring the Job at this site.	>=0; [0 secs]
DEPOT_DELAY		CARDINAL	Delay associated with transferring the goods for this **Stop** at the Depot.	>=0; [0 secs]
WEIGHT		DOUBLE	Define the nett. quantity of goods transferred **into** the Resource. Positive values indicate a pick-up, negative a drop. If the total weight/volume in a Job does not equal 0, the Job must have a Depot to supply/accept the difference.	[0]
VOLUME		DOUBLE		[0]

Stop Packet (Continued) (Sheet 3 of 8)

Field Name	M	Format	Description	Range
SIZE		DOUBLE	Defines the “size” (length, height, weight) of the goods transferred at this Stop. Unlike “weight/volume”, “size” is not cumulative. It is tested against Shift.MAXSIZE to establish compatibility.	>=0; [0]
RELSPEED		DOUBLE	Relative speed in the area. Used to calculate travel-time from the nearest known point on the map to the site.	>=0.01, <=100; [1.0]
ZONES		IDENT	List of zones the Stop is associated with.	[ALL]
J_PKUP		IDENT	List of Job-attributes associated with the Job. The Resource visiting this Stop is deemed to acquire these attributes at departure from the Stop. The attributes will be associated with the Stop until they are “dropped”, or until the next Stop at the associated Depot.	[NONE]
J_DROP		IDENT	The Resource visiting this Stop drops of goods with the specified attributes. A Distribution Stop is deemed to acquire the attributes at the associated Depot.	[NONE]
R_MUSTHAVE		IDENT	List of Resource-attributes a Resource **must** have to visit the Stop.	[NONE]
R_MUSTNOTHAVE		IDENT	List of Resource-attributes a Resource **must not** have to visit the Stop.	[NONE]
R_PREFHAVE		IDENT	List of Resource-attributes a Resource that visits the Stop **preferably** has.	[NONE]
R_PREFNOTHAVE		IDENT	List of Resource-attributes a Resource that visits the Stop **preferably does not** have.	[NONE]
J_MUSTHAVE		IDENT	List of Job-attributes a Resource **must** have acquired when visiting the Stop	[NONE]
J_MUSTNOTHAVE		IDENT	List of Job-attributes a Resource **must not** have acquired when visiting the Stop.	[NONE]
J_PREFHAVE		IDENT	List of Job-attributes a Resource **preferably has** acquired when visiting the Stop.	[NONE]

Stop Packet (Continued) (Sheet 4 of 8)

Field Name	M	Format	Description	Range
J_PREFNOTHAVE		IDENT	List of Job-attributes a Resource **preferably has not** acquired when visiting the Stop.	[NONE]
SINGLE_LINK		TOKEN	If **any** Stop of a Job has this field set, the Stops comprising the Job must be visited in a row, without other Stops in between.	YES,NO [NO]
SINGLE_ITEM		TOKEN	If YES, the Resource must not have any load on board when making this Stop. Usually used in conjunction with SINGLE_LINK to pick-up and deliver goods that may not come into contact with other goods.	YES,NO [NO]
LIFE_SPAN		HHMM	Is the maximum period of elapsed time between departure from this Stop to arrival at the last Stop of the same JOB. (product life-span)	hhmm [] **Note:**Default is blank to indicate "no limit".
VALUE		DOUBLE	Value of this Stop with respect to load-levelling.	>=0,<=1e6; [1.0]
VIEW_ATTRIBUTES		IDENT	Attributes determining which Client has this Stop in its “view”.	[ALL]
SITE_SEPARATION_ID		IDENT	Identifies Stops which must be separated by a separation interval	[IDENT]
SITE_SEPARATION_TIME		HHMM	Is the time interval between the execution of consecutive Stops belonging to the same site.	[2400]
PRODUCT_ID		IDENT	Identifies product associated with Stop.	
PROD_UNITS		CARDINAL	Number of units of associated product Stop contains.	>=0,<=1e6 [0]
SERVICE_COST_MULTIPLIER		DOUBLE	Multiplier cost for Distribution, Collection and Depot Stops with drop_service or pickup_service cost attached.	>=0, <=1e6 [1.0]
MIN_JOB_DELAY		HHMM	The total Job Delay (of all Stops at a Site) should be greater than or equal to this period of time.	>=0; [0 secs]
MAX_JOB_DELAY		HHMM	The total Job Delay (of all Stops at a Site) should be lesser than or equal to this period of time.	>=0; [2400 hours]
ORDER_ID		IDENT	Identifies the Order the Stop belongs to, if any.	[JOB]
PRIMARY_STOP_ID		IDENT	Uniquely identifies primary stop for the order.	

Stop Packet (Continued) (Sheet 5 of 8)

Field Name	M	Format	Description	Range
MIN_OFFSET		HHMM	Minimum period of time to offset from the Primary’s stop time window.	hhmm [0]
MAX_OFFSET		HHMM	Maximum period of time to offset from the Primary’s stop time window. **Note:** MAX_OFFSET should be greater than or equal to MIN_OFFSET	hhmm [MIN_OFFSET]
CAPACITY_ID		IDENT	Defines the type of capacity this stop consumes. If none specified, capacity_id will not match that of any other stop or shift.	
SLA_PRIORITY		DOUBLE	Specifies the fixed cost that will be applied for doing the Job outside its SLA_FLEXIBILITY period.	>=0.001, <=1e6 [1.0]
SHIFT_PROMOTION		DOUBLE	Specifies the multiplier using which the shift-promotion cost for the stop can be increased/decreased.	>=0.0, <=1e6 [1.0]
MATCH_TYPE		CARDINAL	Specifies the geocoder status that will then be displayed on the plannerClient.	>=0, <=9 [0]
BIND_MODE		TOKEN	Determines if a Bound Job can be assigned to the same Shift as the Job to which it is bound. By default all bound jobs are scheduled on separate shifts.	SEPARATE_SHIFT, ANY_SHIFT [SEPARATE_SHIFT]
R_SKILLS		STRING	Defines the Skill that the crew should have to service this Stop. Some jobs may require or prefer certain count or minimum proficiency of the skill.	[NONE]
R_CREW_SIZE		CARDINAL	Defines the minimum number of Crews members that the Shift should have to be able to service the Stop. **For example:** the Job is too heavy to be done by a single crew-member.	>=1;[1]
FIXED_RSRC		STRING	Restricts Scheduler (***but not the Allocator***) from associating the Stop with any but the specified Resource	
spare		STRING	May be used to extend Stop without causing incompatibility with previous versions	

Stop Packet (Continued) (Sheet 6 of 8)

Field Name	M	Format	Description	Range
Customer_ID		STRING		Not used by Oracle Real-Time Scheduler. For display only.
Customer_Name		STRING	The formal customer name	
Street_No		STRING	e.g. 12, second floor	
Street_Name		STRING	e.g. Bowen Grove	
Postcode		STRING	e.g. KY16 9NS	
Location		STRING	e.g. St. Andrews, FIFE	
State		STRING	e.g. Victoria	
Country		STRING	e.g. United Kingdom	
Contact_Name		STRING	e.g. Mr. Jones	
Contact_Phone		STRING	e.g. 13123 12313	
Product_ID		STRING		
Product_Type		STRING	e.g. Furniture	
Product_Description		STRING	e.g. Red sofa bed	
Due_Date		STRING		
Target_Date		STRING		

Stop Packet (Continued) (Sheet 7 of 8)

Field Name	M	Format	Description	Range
Expected_in_Date		STRING	This may be the same as arrdate in Stop planning packet	Not used by Oracle Real-Time Scheduler. For display only.
Cust_Service_ Level		STRING		
Priority		STRING		
Pricing_Information		STRING		
Delivery_Instructions		STRING		
Delivery_Conditions		STRING		
PickUp_Instructions		STRING		
PickUp_Conditions		STRING		
COMMENT		STRING	Comments, e.g. “Beware the cheetah”	
Job_Site		STRING		
Job_Customer_ID		STRING		
Job_Customer_Name		STRING		
Job_Caller_Name		STRING		
Job_Caller_Phone		STRING		
Job_Ref1		STRING		
Job_Ref2		STRING		
Job_Total_Pieces		STRING		
Job_Total_Weight		STRING		
Job_Dim_Weight		STRING		
Job_Master_Waybill		STRING		
Job_House_Waybill		STRING		
Stop_Address_Type		STRING		
Stop_Type		STRING		
Job_Service		STRING		
Job_Sub_Service		STRING		
Job_Revenue_Amount		STRING		
Job_Description		STRING		
Job_Type		STRING		
Job_Class		STRING		
Cust_Details_1		STRING		

Stop Packet (Continued) (Sheet 8 of 8)

Field Name	M	Format	Description	Range
Cust_Details_2		STRING		Not used by Oracle Real-Time Scheduler. For display only.
Cust_Details_3		STRING		
Cust_Details_4		STRING		
Cust_Details_5		STRING		
Cust_Details_6		STRING		
Cust_Details_7		STRING		
Cust_Details_8		STRING		
Cust_Details_9		STRING		
Cust_Details_10		STRING		

7.9.6 Stop Access Time Window

Time-windows for Stops are classified as Access Time Windows (defined by STOP_TW packets) and Service Time Windows (defined by STOP_SLA_TW packets). It is mandatory for a valid Stop to at least have an AccessTW. The Stop cannot be serviced outside the AccessTW.

Time-windows for the Stop are defined by STOP_TW packets. In addition to the window duration itself, each STOP_TW also contains a WINDOW_COST and a LATE_COST field.

Stop AccessTW Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Defines a Stop time-window.	STOP_TW
IDENT	X	IDENT	Uniquely identifies the window.	
STOP	X	IDENT	The Stop to which the window is assigned.	
STATUS		TOKEN	Indicates whether the window may be used or not.	[ACTIVE] INACTIVE
START_DATE		DATE	Time-window during which a Resource may visit the Stop. If not supplied, the referenced window will be deleted.The Stop Time Window Date on which the Stop can be accessed	yyyymmdd
START_TIME		HHMM	The Time of day, on or after which the Stop can be accessed	hhmm If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
DURATION		HHMM	Period of time from the start of its Time Window, during which the Stop will be active	hhmm
WINDOW_COST		DOUBLE	Relative cost to visit within the window.	>=0; [0.0]
LATE_COST		DOUBLE	Relative cost arriving late.	>=0.001,<=1e10; [1.0]
COMMENT		STRING	Comments, e.g. “Beware the cheetah”	Not used by Oracle Real-Time Scheduler. For display only.

Stop AccessTW Packet

Field Name	M	Format	Description	Range
spare		STRING	May be used to extend STOP_TW without causing incompatibility with previous versions	
spare		STRING		
spare		STRING		
spare		STRING		
spare		STRING		
spare		STRING		

7.9.7 Stop Service Time Window

Stop Service Time Window specifies the preferred agreement time for servicing the Stop. Also see section 7.9.6 “Stop Access Time Window” on page 7 - 68.

Oracle Real-Time Scheduler will preferably try to schedule the Job within the SLA Time Window or else incur a cost. The SLA_FLEXIBILITY field specifies the time period before and after the SLA Time Window outside which when the Job is done, a fixed predefined cost is applied.

Time-window during which a Resource may visit the Stop. If not supplied, the referenced window will be deleted.

Stop Service TW Packet (Sheet 1 of 2)

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Defines the Stop’s Service Level Access time window.	STOP_SLA_TW
IDENT	X	IDENT	Uniquely identifies the window.	
STOP	X	IDENT	The Stop to which the window is assigned.	
STATUS		TOKEN	Indicates whether the window may be used or not.	[ACTIVE] INACTIVE
START_DATE		DATE	Date on which the Stop SLA Time Window can be accessed.	yyyymmdd
START_TIME		HHMM	Time of day when the Stop’s SLA Time Window will begin	hhmm If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
DURATION		HHMM	Period of time from the start of its SLA Time Window, when the Stop can be accessed.	hhmm
SLA_FLEXIBILITY		HHMM	Defines the period of time before and after SLA Time Window such that, arriving earlier/later than this time will compulsorily apply a fixed maximum cost.	hhmm
COMMENT		STRING	Comments, e.g. “Beware the cheetah”	Not used by Oracle Real-Time Scheduler. For display only.

Stop Service TW Packet (Continued) (Sheet 2 of 2)

Field Name	M	Format	Description	Range
spare		STRING	May be used to extend STOP_TW without causing incompatibility with previous versions	
spare		STRING		
spare		STRING		
spare		STRING		
spare		STRING		
spare		STRING		

7.9.8 Stop Allocation Priority Configuration

The STOP_AP packet configures the Allocation Priority data specific to the Stop. Stops with "zero" allocation priority will be at the bottom of the list of Stops to be Allocated. With the progress of time SmAuto will internally assess the Stop's allocation status in relation to it's Access and Service Time Windows and escalate it's Allocation Priority.

Note: As the application of Allocation Priority depends on the Stop's Time Windows, the STOP_AP packet for a Stop must be sent after its STOP_TW packet(s).

Stop Allocation Priority Configuration Packet (Sheet 1 of 2)

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Stop-specific Allocation Priority configuration.	STOP_AP
STOP	X	IDENT	The ident of the Stop to which the Allocation Priority configuration is to be applied.	
AP_MODE		TOKEN	Determines how the next six arguments are interpreted: MANDATORY: The stop is mandatory, all other AP fields are ignored. FLAT: The stop is optional, with a flat allocation priority of value AP_STARTVAL. All other AP fields are ignored. FOREVER: The stop is optional with a flat ap-value of AP_STARTVAL up to AP_STARTOFFSET, then linearly increasing to reach AP_ENDVAL at AP_ENDOFFSET, flat thereafter. REVERT: As FOREVER, except that the stop becomes MANDATORY after now reaches AP_ENDOFFSET.	[MANDATORY]' FLAT, FOREVER REVERT
AP_STARTMODE		TOKEN	Determines whether AP_STARTOFFSET is an offset from the beginning of the time-windows, the end or REFDATE. Mandatory if AP_MODE is FOREVER or REVERT.	TWSTART, TWEND, REFDATE
AP_STARTOFFSET		DOUBLE	Allocation-priority start of curve offset from either the beginning of the first window, or the end of the last, depending on the value of AP_STARTMODE. Mandatory if AP_MODE is FOREVER or REVERT	-400.0, 400.0 days [0]

Stop Allocation Priority Configuration Packet (Continued) (Sheet 2 of 2)

Field Name	M	Format	Description	Range
AP_STARTVAL		DOUBLE	Allocation-priority strength until now reaches AP_STARTOFFSET. Mandatory if AP_MODE is FLAT, FOREVER or REVERT	>=0.0
AP_ENDMODE		TOKEN	Determines whether AP_ENDOFFSET is an offset from the beginning of the time-windows, the end or REFDATE. Mandatory if AP_MODE is FOREVER or REVERT.	TWSTART, TWEND, REFDATE
AP_ENDOFFSET		DOUBLE	The Allocation-priority strength rises from AP_STARTVAL to AP_ENDVAL in the interval AP_STARTOFFSET-AP_ENDOFFSET. Beyond AP_ENDOFFSET the Job is mandatory. Mandatory if AP_MODE is FOREVER or REVERT	-400.0, 400.0 days [0]
AP_ENDVAL		DOUBLE	Allocation-priority strength when now is AP_ENDOFFSET. Mandatory if AP_MODE is FOREVER or REVERT.	>STARTVAL
AP_REFDATE		DATE	Date from which offsets are calculated if their mode is REFDATE.	[TODAY]
AP_REFTIME		HHMM	Time of day from when, offsets are calculated if their mode is REFDATE.	If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
spare		STRING	May be used to extend STOP_TW without causing incompatibility with previous versions	
spare		STRING		
spare		STRING		

Oracle Real-Time Scheduler recognises the following Stop Types

Token	Description
LOGON	Represents the start of a Shift. Created and maintained within SmAuto.
LOGOFF	Represents the end of a Shift. Created and maintained within SmAuto.

Oracle Real-Time Scheduler recognises the following Stop Types

PTP	A Stop which is part of a Job not involving a Stop at Depot. PTP stops are internally referred by SmAuto as type STOP.
POU	A Stop which represents a period of unavailability for a Shift. POU stops are ALLOCATED to Shifts and fixed to Resource and Haul.
PKUP	A Stop which picks up goods to be delivered to a Depot.
DROP	A Stop which delivers goods picked up at a Depot.
DIST	Represents a Stop at Depot from where goods are picked up for Distribution by DROP Stops.
COLL	Represents a Stop at Depot where goods from PKUP Stops are dropped.
BREAK	Represents a driver’s Break. Created and maintained within Scheduler.

7.9.9 Slot

The SLOT packet creates a new time slots or replaces an existing one. Time slots will be organized in groups. Whenever a Slot request is made, the SLOT_GROUP field is used to identify the set of Slots from which a selection is to be returned.

Slots can be modified by resending the SLOT packet with updated values.

Slot Generation Packet (Sheet 1 of 2)

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Add/replace a Slot.	SLOT
IDENT	X	IDENT	Identifies the Slot for further manipulation. The ident must be unique within the assigned SLOT_GROUP.	
SLOT_GROUP	X	IDENT	Identifies the group this slot belongs to	
START_DATE	X	DATE	Date on which the Slot is available.	yyyymmdd
START_TIME		HHMM	Time of day when the Slot’s Time Window will begin	hhmm If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
DURATION		HHMM	Period of time from the start of its Time Window, that the Slot will be available.	hhmm

Slot Generation Packet (Continued) (Sheet 2 of 2)

Field Name	M	Format	Description	Range
spare		STRING	May be used to extend SLOT without causing incompatibility with previous versions	
spare				
spare				
spare				
spare				
spare				

7.9.10 Resource

The RSRC packet defines the initial position of a Resource.

Resource Packet (Sheet 1 of 3)

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Defines a Resource.	RSRC
IDENT	X	IDENT	Uniquely identifies the Resource.	
START_LAT	X	DOUBLE	***Initial*** position of the Resource. Accurate positioning needs 6 significant digits after the decimal.	
START_LON				
START_DATE		DATE	***Initial*** date and time of day at which the Resource was at specified position. Either both or none must be present.	yyyymmdd
START_TIME		HHMM		hhmm [NOW] If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
RSRC_COST		DOUBLE	Relative cost for activating the Resource (= at least one active Shift).	[1.0]

Resource Packet (Continued) (Sheet 2 of 3)

Field Name	M	Format	Description	Range
spare		STRING	May be used to extend RSRC without causing incompatibility with previous versions	
spare		STRING		
spare		STRING		
spare		STRING		
spare		STRING		
spare		STRING		
Vehicle_ID		STRING		Not used by Oracle Real-Time Scheduler. For display only.
Vehicle_Description		STRING		
Special_Conditions		STRING		
License_Number		STRING		
Max_Speed		STRING		
Nominal_Logon_Street_Num		STRING	This is the “default” logon street number of the Resource	
Nominal_Logon_Street_Name		STRING	This is the “default” logon street name number of the Resource	
Nominal_Logon_Location		STRING	This is the “default” logon city number of the Resource	
Nominal_Logon_State		STRING	This is the “default” logon state number of the Resource	
Nominal_Logon_Postcode		STRING	This is the “default” logon postcode number of the Resource	

Resource Packet (Continued) (Sheet 3 of 3)

Field Name	M	Format	Description	Range
Nominal_Logoff_Street_Num		STRING	This is the “default” logoff street number of the Resource	Not used by Oracle Real-Time Scheduler. For display only.
Nominal_Logoff_Street_Name		STRING	This is the “default” logoff street name number of the Resource	
Nominal_Logoff_Location		STRING	This is the “default” logoff city number of the Resource	
Nominal_Logoff_State		STRING	This is the “default” logoff state number of the Resource	
Nominal_Logoff_Postcode		STRING	This is the “default” logoff postcode number of the Resource	
Driver_Contact_Details		STRING		
Designated_Depot		STRING		
Cost_Per_Distance_Unit		STRING	In absolute monetary values	
Cost_Per_Time_Unit		STRING	In absolute monetary values	
Designated_Area		STRING	The geographic area the Resource is associated with	
Insurance_Details		STRING		
COMMENT		STRING	Comments, e.g. “Short trips only: freezer broken”	

7.9.11 Shift

The Shift packet defines a single Shift for a Resource. Most of the Resource characteristics are built into the Shift and hence allowing for different configurations on different days.

Shift Packet (Sheet 1 of 7)

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Defines a Shift.	SHIFT
IDENT	X	IDENT	Uniquely identifies the Shift.	
RSRC	X	IDENT	Uniquely identifies the Resource the Shift is associated with.	
MDT		IDENT	Uniquely identifies the mobile data terminal the Shift is associated with.	
TYPE		TOKEN	See **“Oracle Real-Time Scheduler recognises the following Shift Types” on page 7 - 84** for a listing of **ShiftType**s.	[STANDARD]
START_LAT	X	DOUBLE	Log-on position for the Shift. Accurate positioning needs 6 significant digits after the decimal.	
START_LON				
END_LAT		DOUBLE	Log-off position of the Shift. Accurate positioning needs 6 significant digits after the decimal.	[START_LAT/ LON}
END_LON				
START_DATE	X	DATE	START_DATE is the Date that the Resource will be available for Shift work. START_TIME is the time of day on or after which the Shift work can start. DURATION is the period of time that the Resource will be available for Shift work. A Resource must commence the Shift after the start of the time-window and finish before the end. **Note:** Scheduler assumes that Shifts for the same Resource will be started in order of their start-time. Therefore, the start-time of two Shifts for the same Resource must not be identical.	yyyymmdd
START_TIME		HHMM		hhmm If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
DURATION		HHMM		hhmm
MAX_SHIFT		HHMM	Maximum length of the Shift. The length of a Shift is calculated as the time period between estimated/ actual departure from log-on position and estimated arrival at log-off position.	[END-START]

Shift Packet (Continued) (Sheet 2 of 7)

Field Name	M	Format	Description	Range
MAX_HAUL		HHMM	Is the maximum length in time, for the Haul at the end of this Shift. Limits the sum of the Shift-lengths from the first Shift ***of the Haul*** to this one inclusive.	[unlimited]
MAX_RUNLENGTH		HHMM	Maximum length in time, for any run which starts within the Shift. **Note:** The MAX_RUNLENGTH field will be applied to any run that starts within the Shift for which it has been set. A cost is applied to every second that MAX_RUNLENGTH exceeds that of the estimated run-duration.	>=1
REST_PERIOD		HHMM	Minimum length in time of the rest-period between the end of this Shift and the start of the next. LONGHAUL only.	hhmm
SHIFT_COST		DOUBLE	Relative cost for activating the Shift.	>=0.001; [1.0]
OVERTIME_COST		DOUBLE	Relative cost for exceeding MAX_SHIFT or START_TIME + DURATION.	>=0.001,<=1e 10; [1.0]
LONGHAUL_COST		DOUBLE	Relative LONGHAUL cost. (See CostControl)	>=0; [1.0]
MAXWEIGHT		DOUBLE	Maximum weight the Resource can carry.	[unlimited]
MAXVOLUME		DOUBLE	Maximum volume the Resource can carry.	[unlimited]
MAXSIZE		DOUBLE	Limits the size (e.g. length, height, weight) of an individual parcel. May be used to restrict parcels that are too long, too heavy (for the driver to carry) etc.	[unlimited]
DIST_COST		DOUBLE	Relative travel distance cost.	>=0.01, <=100 [1.0]
TIME_COST		DOUBLE	Relative travel time cost.	>=0.01, <=100 [1.0]
RELSPEED		DOUBLE	Relative speed of the Resource.	>0.01, <=100; [1.0]
ZONE_MUST		IDENT	List of zones the Resource can work during this Shift.	[ALL]
ZONE_PREF		IDENT	List of zones the Resource preferably works during this Shift.	[ALL]

Shift Packet (Continued) (Sheet 3 of 7)

Field Name	M	Format	Description	Range
ATTRIBUTES		IDENT	List of Resource-attributes for the duration of the Shift.	[NONE]
DEPOT		IDENT	Comma separated list of Depots where goods may be collected from or dropped off at. **Note:** If specified, assigns Shifts to the Depot.	[NONE]
TARGET_VALUE		DOUBLE	Desired slice of "average total Stop-value per Shift". **Note:** "average total Stop-value" is calculated over active Shifts. TARGET_VALUE cannot be used to force SmAuto to use all available Shifts.	>=0,<=1e6; [1.0]
VIEW_ATTRIBUTES		IDENT	Attributes determining which Client has this Shift in its "view".	[ALL]
LOGON_DELAY		HHMM	Time Duration for logon Stop.	hhmm
LOGOFF_DELAY		HHMM	Time Duration for logoff Stop.	hhmm
SKILLS		STRING	Defines the Skills belonging to the Shift's crew members.	[NONE]
FIXED_LOGON		TOKEN	Determines the Shift logon time within the Shift's time window. FIXED_LOGON=YES forces Scheduler to fix the Shift's logon at the start of its time window.	YES, NO, [DEFAULT]
AUTODIRECT		TOKEN	Determines the jobs on the shift can be Autodirected.	YES, NO, [YES]
JOB_HORIZON		CARDINAL	The number of jobs to be Autodirected in advance.	>=1;[1]
AUTO_ENROUTE		TOKEN	Sets Autodirected job status to ENROUTE if value set to YES or to DESPATCHED if set to NO. AUTO_ENROUTE can be set to NO when either despatch messages take some time to reach the driver or multiple Stops are despatched at a time and the driver has the option to change the order in which the Stops are to be activated.	YES, NO, [YES]

Shift Packet (Continued) (Sheet 4 of 7)

Field Name	M	Format	Description	Range
COST_WAIT_SHIFT		TOKEN	Costs idle-time of a driver weighted according to the "time from now" factor. No cost is applied if the idleness accrues further than WAIT_HORIZON time interval (set in cost controls, 2 hours by default) and within this horizon the applied cost is stronger if the idleness is closer to the current time.	YES, NO, [YES]
AUTO_GO_HOME		TOKEN	Defines whether automatic go home operations are allowed for the shift.	YES, [NO]
TIME_GO_HOME		HHMM	Defines the period of time driver is kept idle at last completed stop before automatic directing to logoff.	hhmm[] note: default is blank to indicate direct home on stop completion.
CAPACITY_TYPE		IDENT	Defines the type of capacity to be reserved. If not specified, no capacity is reserved.	
RESERVE_CAPACITY		DOUBLE	The amount of capacity to reserve.If not specified, no capacity is reserved.	>=0.0, <=1.0 [0]
CAPACITY_LEADTIME		HHMM	Is a time period. Reserved capacity will be released when the wall-time reaches (START_TIME - CAPACITY_LEADTIME). If not specified, reserved capacity is never released.	
EOTT_ON_START		HHMM	Defines the maximum period of time that the engineer travels from home to first job "at his own expense".	hhmm[]
EOTT_ON_FINISH		HHMM	Defines the maximum period of time that the engineer travels from last job to home "at his own expense".	hhmm[]
REL_EFFICIENCY		DOUBLE	Extends or shortens the Stop’s JOB_DELAY time representing the efficiency of a shift to complete a job.	>=0.01, <=100 [1.0]
JOBS_LIMIT_VALUE		DOUBLE	Specifies the maximum number of jobs that can be done by a shift. Job’s in this context are "Customer" Jobs only, and exclude depot/pou/collection jobs. "0" value indicates that shift has no jobs limit.	>=0.0, <=2e+9 [0.0]
DW_AREA_LAT		DOUBLE	Specifies the Latitude for the dynamic working area.	

Shift Packet (Continued) (Sheet 5 of 7)

Field Name	M	Format	Description	Range
DW_AREA_LON		DOUBLE	Specifies the Longitude for the dynamic working area.	
RELATIVE_DW_AREA		DOUBLE	Specifies the relative factor for the dynamic working area.	>=0.001, <=1000 [1]
SEQ_LOCKING		TOKEN	Flag to determine if any despatched Stop should automatically be sequence locked (LOCK_SEQ) or locked by order (LOCK_ORDER) on the Shift. When unlocking is required, Planner automatically changes the sequence of affected Stops while retaining the stop Statuses. If LOCK_SEQ is On, Smauto will not introduce new stops before despatched Stops, nor change the order of despatched Stops. If LOCK_ORDER is ON, the order of despatched stops will not be affected but new stops may be introduced before them.	NO_LOCKING, LOCK_SEQ, LOCK_ORDER [NO_LOCKING]
AUTO_UNDESP		TOKEN	If set to YES, on encountering an out-of-sequence event Planner would change the status of any despatched Stops outside the Despatch Horizon to UNDESPATCHED. (Example for an out-of-sequence event: Despatch Horizon is 3 and 3 Stops A, B & C have been despatched to a single Crew. ASSIGNED Stop X is despatched and unexpectedly enrouted, causing Stop C to be outside the Despatch Horizon.) The UNDESPATCHED status merely acts as a trigger for the Host to inform the Crew and reset the Stop status to ASSIGNED or ALLOCATED.	YES, NO [NO]
CREW_SIZE		CARDINAL	Defines the number of Crew members that are available on the Shift.	>=1; [1]

Shift Packet (Continued) (Sheet 6 of 7)

Field Name	M	Format	Description	Range
Logon_Street_Num		STRING		These fields are for display purpose only and will not be used by Oracle Real-Time Scheduler for scheduling.
Logon_Street_Name		STRING		
Logon_Location		STRING		
Logon_State		STRING		
Logon_Postcode		STRING		
Logon_Time		STRING		
Logon_Date		STRING		
Logoff_Street_Num		STRING		
Logoff_Street_Name		STRING		
Logoff_Location		STRING		
Logoff_State		STRING		
Logoff_Postcode		STRING		
Logoff_Time		STRING		
Logoff _Date		STRING		
Current_Location		STRING		
Current_Time		STRING		
Current_Date		STRING		
Driver_ID		STRING		
Driver_Name		STRING		
Co_Driver_Name		STRING		
Contact_Phone		STRING		
MDT_Status		STRING		
MDT_Number		STRING		
Trailer_ID		STRING		
Trailer_Description		STRING		
Equipment		STRING	Equipment carried by the Resource for this Shift	

Shift Packet (Continued) (Sheet 7 of 7)

Field Name	M	Format	Description	Range
Special_Conditions		STRING		These fields are for display purpose only and will not be used by Oracle Real-Time Scheduler for scheduling.
Special_Instructions		STRING	i.e. “Left rear Break light is broken - please fix”	
Emergency_Contact_Name		STRING		
Emergency_Contact_Phone		STRING		
COMMENT		STRING	Comments, e.g. “Short trips only: freezer broken”	

Oracle Real-Time Scheduler recognises the following Shift Types

Token	Description
STANDARD	The Shift commences at the log-on position and terminates at the log-off position. No goods remain on the Resource at log-off.
ONE_WAY	The Shift terminates at the last Stop in the tour. No goods remain on the Resource. Any subsequent Shift commences at the specified log-on position.
MANYSHIFT	As “standard”, but the start of the next Shift depends on log-off time of this Shift. Incurs a high cost when goods/jobs are carried to the next shift.
MULTISHIFT	As “manyShift”, but additionally goods/Jobs may be carried into the next Shift.
LONGHAUL	The Shift does not necessarily terminate at the log-off position. Actual log-off position and any goods on board will be “carried” forward to the start of the next Shift.

7.9.12 Break

Break packets define Breaks for each Shift. Break’s status and corresponding times (if it is necessary) will be send in STOP_PARAM packet

Break Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Defines a single Break for a specific Shift.	Break
IDENT	X	IDENT	Uniquely identifies the Break.	
SHIFT	X	IDENT	The Shift to which the Break is assigned.	
BREAK_ORDER		CARDINAL	Sequence # for Breaks of a Shift. The sequence number determines the order in which Breaks are taken.	>=1; [1]
AUTOCOMPL_MODE		TOKEN	When REALTIME is set to AUTO_COMPLETE, this Break will be automatically “completed” as scheduled.	[YES],NO
OFFSET	X	HHMM	OFFSET is the time period from the previous event (see TYPE) that this Break should start. Duration is the time period from OFFSET within which the Break should be started.	hhmm
DURATION		HHMM		hhmm
DELAY	X	HHMM	Length of the Break period.	hhmm
TYPE	X	TOKEN	Determines how OFFSET is interpreted. See Break-Types table.	Shift, LOGON, ELAPSED, DRIVE
RESTRICT		TOKEN	Defines any restrictions to the placing of the break. When set to ATDEPOT, the Break can be scheduled at or after arrival at a Store only. Otherwise all rules governing the scheduling of Breaks apply. **Note:** ATDEPOT restriction is not applicable for DRIVE Breaks.	[NONE], ATDEPOT

Break Packet

Field Name	M	Format	Description	Range
EXTEND		HHMM	Extends the range of allowable times to schedule a break with the specified number of minutes. The extra time will be added *before* the time-window specified by OFFSET and DURATION. If no value is specified, it will be defaulted from the breakEarlyExtend command-line argument. See accompanying Oracle Real-Time Scheduler "System Manual" for further details on command line arguments.	hhmm
spare		STRING	May be used to extend Break without causing incompatibility with previous versions	
spare				
spare				
spare				
spare				
spare				

Oracle Real-Time Scheduler recognises the following Break-Types

Token	Description
SHIFT	Offset is calculated in elapsed time from the earliest possible start of the Shift.
LOGON	Offset is calculated in elapsed time from the predicted or actual time of logon.
ELAPSED	Offset is calculated in elapsed time from the predicted or actual start of the previous Break or logon.
DRIVE	Offset is calculated in drive time from the predicted or actual start of the previous Break or logon.

7.10 Oracle Real-Time Scheduler Object Delete Packets

This section describes packets used by Oracle Real-Time Scheduler to delete Objects from its store

Object Delete Packets

Packet Type	Description
DEPOT_DEL	Deletes a Depot.
DEPOT_TW_DEL	Removes a Depot Access time-window.
DEPOT_SLA_TW_DEL	Removes a Depot Service time-window.
RSRC_DEL	Deletes a Resource.
SHIFT_DEL	Deletes a Shift.
STOP_DEL	Deletes a Stop.
STOP_TW_DEL	Removes a Stop Access time-window.
STOP_SLA_TW_DEL	Removes a Stop Service time-window.
BREAK_DEL	Deletes a Break.
PROD_DEPOT_DEL	Deletes a Product-Depot relationship.
SLOT_DEL	Deletes a slot

7.10.1 Deleting Stop, Depot, Shift, Resource, Break and Time Windows

Stop/StopTW, Depot/DepotTW, Shift, Resource Break Delete Packets

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Deletes a Stop. DEPOT_STOPs cannot be deleted explicitly. The system creates and deletes them as necessary.	STOP_DEL
			Deletes a DEPOT.	DEPOT_DEL
			Deletes a Shift. Any Stops assigned to the Shift will be un-allocated. Any Breaks will be deleted.	SHIFT_DEL
			Deletes a Resource and all associated Shifts and Breaks.	RSRC_DEL
			Deletes a Break.	BREAK_DEL
			Deletes a Stop Access time-window.	STOP_TW_DEL
			Deletes a Stop Service time-window.	STOP_SLA_TW_DEL
			Deletes a Depot Access time-window.	DEPOT_TW_DEL
			Deletes a Depot Service time-window.	DEPOT_SLA_TW_DEL
IDENT	X	IDENT	Identifies object to operate on.	

7.10.2 Deleting Product Depots

Product Depot Delete Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Remove product-Depot relation	PROD_DEPOT_DEL
PRODUCT_ID	X	IDENT	Product to be removed from the Depot.	
DEPOT_ID	X	IDENT	Depot this product has to be removed from	

7.10.3 Deleting time-slots

The SLOT_DEL packet deletes a single slot at a time irrespective to, the Slot belonging to a Slot Group or not.

Deleting Slots

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Find and return time slots.	SLOT_DEL
IDENT	X	IDENT	Identifies a single Slot to be deleted.	

7.11 PLAN AND PARAMETER PACKETS

Plan-packets describe how the various objects are organized into a plan. The totality of al PLAN_* packets determine a Plan, i.e. saving and reloading all PLAN_* packets results in an identical plan.

Plan Packets

Packet Type	Description
PLAN_STOP	Advises the current placing of a Stop within the “plan”.
PLAN_SHIFT	Current status of a Shift.
PLAN_RSRC	Advises the estimated current position of a Resource.
PLAN_DEPOT	Advises the current placing of a Depot within the “plan”.
PLAN_DEPOT_TW	Depot time-window specific information.

Plan-parameter packets are identical to “plan” packets, with additional cost and time-information for the benefit of the recipient.

Param Packets

Packet Type	Description
PARAM_STOP	PLAN_STOP and timing info.
PARAM_SHIFT	Provides total cost.
PARAM_RSRC	Advises the estimated current position of a Resource. Provides total cost.
PARAM_DEPOT	PLAN_DEPOT and timing info.
PARAM_DEPOT_TW	Exports remaining capacity information of the Depot.

Note: From the point of view of network-traffic, it would be better if “parameter” packets would just contain parameters, without any PLAN-fields. This however increases the complexity of interaction between a Client and SmAuto.

7.11.1 Resource Plan packet

The PLAN_RSRC packet keeps the recipient informed of any position changes of the Resource.

Resource Planning Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Returns schedule dependent info.	PLAN_RSRC
IDENT	X	IDENT	Resource-id.	
RESUME_DATE		DATE	Date and time of day at which the driver is assumed to resume work, if temporarily delayed. if current time < resume-time status will be changed to DELAYED.	yyyymmdd hhmm If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
RESUME_TIME		HHMM		
LATITUDE	X	DOUBLE	Position of the Resource. Accurate positioning needs 6 significant digits after the decimal.	
LONGITUDE				
DATE		DATE	Date and time of day at which the Resource was at the specified position. Either both or none must be present.	yyyymmdd hhmm [NOW] If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
TIME		HHMM		
spare		STRING	May be used to extend PLAN_RSRC without causing incompatibility with previous versions	
spare				
spare				

7.11.2 Resource Parameter Packet

PARAM_RSRC packet advises Clients of status and current position of a Resource.

Resource Parameter Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Returns schedule dependent info.	PARAM_RSRC
IDENT	X	IDENT	Resource-id.	
Body of the PLAN_RSRC packet.				
STATUS		TOKEN	The status of the Resource, as reported by SmAuto.	ON OFF ARRIVED ONSTOP ONBREAK DELAYED
TIME_TO		HHMMSS	Time/Distance from the current position to the next Stop along the most direct route.	
DIST_TO				
TRAVELDIST	X	DOUBLE	Sum for all Shifts.	
TRAVELTIME	X	HHMMSS		
IDLETIME	X	HHMMSS		
LOADTIME	X	HHMMSS		
DRIVE_DURATION		HHMMSS		
WORK_DURATION		HHMMSS		
ACTIVE_VIEWATTRS		IDENT	Names of all the active Views in which the Rsrc is visible	[ALL]
ACTIVE_DATEATTRS		IDENT	List of dates included in the view. This field may contain both individual dates and date-ranges: 20060110, 20060112-20060114, 20060116-20060118	[ALL]
spare		STRING	May be used to extend PARAM_RSRC without causing incompatibility with previous versions	
spare				
spare				
spare				
spare				
spare				

7.11.3 Depot Planning

Depot Plan Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Returns schedule dependent info.	PLAN_DEPOT
IDENT	X	IDENT	Depot id.	
ERD_DATE		DATE	Earliest Run Departure Time. The earliest time of the day when a run can depart from a Distribution Depot	
ERD_TIME		HHMM		If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
spare		STRING	May be used to extend PLAN_DEPOT without causing incompatibility with previous versions	
spare				
spare				

7.11.4 Depot Parameter Packet

On receipt of the DEPOT_CUTOFF packet, immediately after all the affected runs have been closed, Oracle Real-Time Scheduler will adjust the Earliest Run Departure times for each Depot to be equal to the CUTOFF_DATE/TIME. The change will be exported to the Switch in the PARAM_DEPOT packet. Packets of type PARAM_DEPOT may also be passed to the Switch as part of a DBLoad DATA_START/END sequence. The Switch will pass them on to Scheduler, updating the Earliest Run Departure time applied by Scheduler and Slot-generators.

Depot Parameter Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Depot specific parameters.	PARAM_DEPOT
IDENT		IDENT	Identifies the Depot.	
ERD_DATE	X	DATE	Open runs cannot depart from Depot until this time of the day.	
ERD_TIME		HHMM		If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
spare		STRING	May be used to extend PARAM_DEPOT without causing incompatibility with previous versions	
spare				
spare				
spare				
spare				
spare				
spare				
spare				
spare				

7.11.5 Depot TimeWindow Planning

DepotTW Plan Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Returns schedule dependent info.	PLAN_DEPOT_TW
IDENT	X	IDENT	Depot TW id.	
spare		STRING	May be used to extend PLAN_DEPOT_TW without causing incompatibility with previous versions	
spare				
spare				

7.11.6 Depot TimeWindow Parameter

The PARAM_DEPOT_TW packet is used to export the Remaining Capacity of a Depot during a time-window.

DepotTW Parameter Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Depot specific parameters.	PARAM_DEPOT_TW
IDENT	X	IDENT	Identifies the Depot.	
spare		STRING	May be used to extend PLAN_SHIFT without causing incompatibility with previous versions	
spare				
spare				
REMWEIGHT	X	DOUBLE	Remaining Capacity.	
REMVOLUME	X	DOUBLE		
spare		STRING	May be used to extend PLAN_SHIFT without causing incompatibility with previous versions	
spare				
spare				
spare				
spare				
spare				

7.11.7 Shift Plan packet

The PLAN_SHIFT packet advises the Client of the Shift’s status.

Shift Plan packet (Sheet 1 of 2)

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Returns schedule dependent info.	PLAN_SHIFT
IDENT	X	IDENT	Shift-id.	
STATUS	X	TOKEN		INACTIVE PLANNED CLOSED STARTED STANDBY COMPLETING COMPLETED
STARTED_DATE		DATE	Date and time of day at which the Shift was started/completed. Compulsory when STATUS=STARTED/COMPLETING/ COMPLETED.	yyyymmdd
STARTED_TIME		HHMM		hhmm If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
COMPLETED_DATE		DATE		yyyymmdd
COMPLETED_TIME		HHMM		hhmm If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
MUST_GO_HOME		TOKEN	Flag is set to YES with a manually activated GO_HOME operation. But the GO_HOME instruction is sent to the driver only after all despatch stops in Shift are done. **Note:** SmAuto also sets the GOING_HOME flag to true after sending the GO_HOME instruction to MDT.	YES, [NO]

Shift Plan packet (Continued) (Sheet 2 of 2)

Field Name	M	Format	Description	Range
GO_HOME_TIME		HHMM	Defines the period of time, that the driver is kept idle at last completed stop before being automatically directed to logoff.	hhmm[] note: default is blank to indicate direct home on stop completion.
GOING_HOME		TOKEN	Defines whether driver has been directed to logoff. **Note:** This flag will be set either as a result of the AUTO_GO_HOME or MUST_GO_HOME instruction.	YES, [NO]
spare		STRING	May be used to extend PLAN_SHIFT without causing incompatibility with previous versions	
spare				
spare				

7.11.8 Shift Parameter Packet

PARAM_SHIFT packet advises Clients of distance and time associated with a Shift. Log on/off position/time will be supplied in the PLAN_STOP packets.

Shift Parameter Packet (Sheet 1 of 3)

Field Name	M	Format	Format	Range
PACKET_NAME	X	TOKEN	Returns schedule dependent info.	PARAM_SHIFT
IDENT	X	IDENT	Shift-id.	
Body of the PLAN_SHIFT packet.				
HAUL		IDENT	Id of the Haul of 2 or more Haul this Shift belongs to, if any.	

Shift Parameter Packet (Continued) (Sheet 2 of 3)

Field Name	M	Format	Format	Range
START_DRIVING_DATE		DATE	Defines date and time of the day that the driver logs on and starts driving.	yyyymmdd
START_DRIVING_TIME		HHMM		hhmm If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
DRIVE_DURATION		HHMMSS	Defines the period of shift time, that was spent driving.	hhmmss
START_WORK_DATE		DATE	Defines date and time of day that the driver starts working for the company.	yyyymmdd
START_WORK_TIME		HHMM		hhmm If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
WORK_DURATION		HHMMSS	Defines the period of time that was spent working on the shift.	hhmmss
TRAVELDIST	X	DOUBLE	Total distance travelled between log-on and log-off for this Shift.	

Shift Parameter Packet (Continued) (Sheet 3 of 3)

Field Name	M	Format	Format	Range
TRAVELTIME	X	HHMMSS	TravelTime is derived on a "run" basis, i.e. It is the sum of all the derived time between departure from LogStops and Depots to the arrival at the next LogStop or Depot within a Shift. When there is a break between the previous stop and the Depot/Log Stop, any waiting-time for the break is not included, as it can be spent at depot/logoff. The latter discourages the time spent "waiting" for a break at depot from being included in the TravelTime calculation. The aim is to determine the time period that the driver is "working" or on break. Time spent at depot, logon, logoff (unless on break) is excluded from the total TravelTime. Any delays at Depot (loading, waiting) is not be included in the TravelTime calculation. Time spent on Breaks at Depot are included. **Note:** The time spent **"on Break"** refers to Breaks actually scheduled or taken, not the total amount of Break-time available	hhmmss
IDLETIME	X	HHMMSS	Total time spent waiting at Stops and Depots only, **excluding any time spent at Break Stops**.	hhmmss
LOADTIME	X	HHMMSS	Total time spent loading/unloading goods.	hhmmss
ACTIVE_DATEATTRS		IDENT	List of dates included in the view. This field may contain both individual dates and date-ranges: 20060110, 20060112-20060114, 20060116-20060118	[ALL]
spare		STRING	May be used to extend PARAM_SHIFT without causing incompatibility with previous versions	
spare				
spare				
spare				
spare				
spare				

Oracle Real-Time Scheduler Shifts have one of the following Status (Sheet 1

Token	Description

Oracle Real-Time Scheduler Shifts have one of the following Status (Sheet 2

INACTIVE	The Shift is not included into the schedule.
PLANNED	The Shift is currently not logged on to, and never has been logged on to. Scheduler can assign work to the Shift.
STANDBY	This Shift is currently not logged on to. Jobs cannot be assigned to this Shift until it is logged on.
CLOSED	As PLANNED, but may not be recommended for additional work.
STARTED	A Resource is currently logged on to the Shift.
COMPLETING	As STARTED, but may not be recommended for additional work.
COMPLETED	A Resource has logged off from the Shift, after completing all Jobs.

7.11.9 Stop Plan Packet

The PLAN_STOP packet advises Clients of the status of a STOP.

Stop Plan Packet (Sheet 1 of 5)

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Returns schedule dependent info.	PLAN_STOP
IDENT	X	IDENT	Identifies the Stop to be updated.	
TYPE	X	TOKEN	Regular Stop.	PKUP, DROP, STOP, POU
			Stop at Depot. **Note:** Note: if the referenced Stop does not exist, SmAuto will *create* an instance.	DIST, COLL
			Shift Stop. May only be used to register status FREE, STARTED/ COMPLETED.	LOGON, LOGOFF
			Break Stop. May only be used to register status FREE,/STARTED/ COMPLETED.	BREAK
STATUS	X	TOKEN	See section 7.11.10.1 “Stop statuses associated with different Stop types” on page 7 - 108.	INACTIVE FREE ASSIGNED ALLOCATED CLOSED DESPATCHE PENDING_DESPATCHED UNDESPATCHED ACKNOWLEDGED ENROUTE ARRIVED POSTPONED, STARTED COMPLETED
SHIFT		IDENT	Shift within which the Stop is visited. If not set, the STOP is un-allocated. Compulsory when status = ASSIGNED - COMPLETED.	
STOPNO		CARDINAL	Determines the position of the Stop, within the Shift. Compulsory when status = ASSIGNED - COMPLETED.	

Stop Plan Packet (Continued) (Sheet 2 of 5)

Field Name	M	Format	Description	Range
DEPOT		IDENT	IDENT of the Depot at which goods for this Stop are loaded/ dropped, or in case of a DIST/ COLL-Stop, the associated Depot. Compulsory when status = CLOSED - COMPLETED.	
DEPOT_STOP		IDENT	IDENT of the Depot-Stop at which the goods for this Stop were loaded/will be unloaded. Compulsory when status = CLOSED - COMPLETED.	
FIXED_DEPOT		IDENT	Restricts Scheduler (***but not the Allocator***) from associating the Stop with any but the specified object(s)	
FIXED_HAUL		IDENT		
FIXED_RSRC		IDENT		
GROUP_STATUS		TOKEN	Indicates the level of grouping See GroupStatus table.	[INDEPENDENT], GROUPED, LINKED
GROUP		IDENT	Group the Stop belongs to.	
GROUP_SEQ		CARDINAL	Sequence number to order Stops within group.	[0]
POSTPONED_DATE		DATE	Date and Time of day to which the Stop is postponed. Must be set if status = POSTPONED.	yyyymmdd
POSTPONED_TIME		HHMM		hhmm If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))

Stop Plan Packet (Continued) (Sheet 3 of 5)

Field Name	M	Format	Description	Range
ARRIVED_DATE		DATE	Date and time of day at which the Stop was arrived at/started/ completed. Provides the dead-reckoning sub-system with a starting-point. Compulsory when STATUS=ARRIVED/STARTED/ COMPLETED.	yyyymmdd
ARRIVED_TIME		HHMM		hhmm If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
STARTED_DATE		DATE		yyyymmdd
STARTED_TIME		HHMM		hhmm If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
COMPLETED_DATE		DATE		yyyymmdd
COMPLETED_TIME		HHMM		hhmm If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
LATITUDE		DOUBLE	Position of the Stop. Accurate positioning needs 6 significant digits after the decimal.	
LONGITUDE				
SUB_WIN_DATE		DATE	Defines a SubWindow date.	

Stop Plan Packet (Continued) (Sheet 4 of 5)

Field Name	M	Format	Description	Range
SUB_WIN_TIME		HHMM	Time of day at which the SubWindow will start.	hhmm If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
SUB_WIN_DURATION		HHMM	Period of time that the SubWindow will exist.	hhmm
DUE_DATE		DATE	Is the ETA (Expected Time of Arrival) AFTER the driver's route has been updated with the ENROUTE status.	yyyymmdd
DUE_TIME		HHMMSS		hhmmss If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)). That is: ((local time - UTC offset) to (local time + UTC offset))
ETA_DATE		DATE	Is the external ETA (Expected Time of Arrival) received either from the STOP_ETA_UPDATE packet or the ETA_FROM_MDT packet.	
ETA_TIME		HHMMSS		If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)). That is: ((local time - UTC offset) to (local time + UTC offset))
COMPAT_RADIUS		DOUBLE	This value ensures that schedules are reproducable. If set, it will override any "radius" that is calculated internally.	

Stop Plan Packet (Continued) (Sheet 5 of 5)

Field Name	M	Format	Description	Range
BIND_MODE		TOKEN	This value determines if the Bound Job will be assigned to the same Shift as the Job to which it is bound. By default all bound jobs are scheduled on separate shifts.	SEPARATE_SHIFT, ANY_SHIFT
spare		STRING	May be used to extend PLAN_STOP without causing incompatibility with previous versions	

7.11.10 Stop Parameter Packet

In addition to the info in the PLAN_STOP packet, the PARAM_STOP packet contains useful derived information, e.g. ETA, load-on-departure etc.

Stop Parameter Packet (Sheet 1 of 4)

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Returns schedule dependent info.	PARAM_STOP
IDENT	X	IDENT	Identifies the Stop to be updated.	
Body of the PLAN_STOP packet.				
ERROR		TOKEN	Displays error messages associated with the Stop if any.	NONE, DISABLED
JOB_STATUS		TOKEN	Displays the status of the Job to which the Stop belongs.	FREE ASSIGNED ALLOCATED DESPATCHED PENDING_DESPATCHED UNDESPATCHED ACK-NOWLEDGED CLOSED STARTED COMPLETING COMPLETED
RUN_STATUS		TOKEN	Displays the status of the Run of which the Stop is a part of.	PLANNED CLOSED STARTED COMPLETING COMPLETED

Stop Parameter Packet (Continued) (Sheet 2 of 4)

Field Name	M	Format	Description	Range
ALLOC_PRIORITY		STRING	Displays the allocation priority assigned to the Stop.	
WINDOW		IDENT	IDENT of the window that SmAuto has selected for the Stop.	
ARRDATE		DATE	Date and time of day at which the Stop was arrived at.	yyyymmdd
ARRTIME		HHMMSS		hhmmss If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)). That is: ((local time - UTC offset) to (local time + UTC offset))
STARTDATE		DATE	Date and time of day at which the Stop was started.	yyyymmdd
STARTTIME		HHMMSS		hhmmss If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)). That is: ((local time - UTC offset) to (local time + UTC offset))

Stop Parameter Packet (Continued) (Sheet 3 of 4)

Field Name	M	Format	Description	Range
DEPDATE		DATE	Date and time of day at which the Stop was completed	yyyymmdd
DEPTIME		HHMMSS		hhmmss If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)). That is: ((local time - UTC offset) to (local time + UTC offset))
LOCATION		STRING	This is the current location of the Resource (the closest map node name is reported here)	
ZONE		IDENT	Zone to which the Stop has been assigned (through AutoZoning).	
DEPVOL		DOUBLE	Volume on board at departure.	
DEPWHT		DOUBLE	Weight on board at departure.	
DEPATTRS		IDENT	JOB_ATTRS on board at departure.	[NONE]
TIME_FROM		HHMMSS	Total Time/Distance from previous Stop along the most direct route. (ignores the current position of the Resource)	
DIST_FROM		DOUBLE		
RSRC		IDENT	Uniquely identifies the Resource the Stop is associated with	
LATE_TIME		DOUBLE	Calculated as MAX(0, ETA-latest time in associated time window) for a Stop	
TIME_ONSITE		DOUBLE	Calculated as ETD - ETA for a Stop	
ACTIVE_VIEWATTRS		IDENT	Names of all the active Views in which the Stop is visible	[ALL]
ACTIVE_DATEATTRS		IDENT	List of dates included in the view. This field may contain both individual dates and date-ranges: 20060110, 20060112-20060114, 20060116-20060118	[ALL]
EST_TIME_FROM		HHMMSS	Displays the estimated travel-time between stops	

Stop Parameter Packet (Continued) (Sheet 4 of 4)

Field Name	M	Format	Description	Range
WARN_TEXT		STRING	Contains the reason, the Job has been disabled.	
spare		STRING	May be used to extend PARAM_STOP without causing incompatibility with previous versions	
spare				
spare				
spare				
spare				

7.11.10.1 Stop statuses associated with different Stop types

There are four types of Stops:

- Log-Stops, which mark the start and end of a Shift.
- Depot-Stops, which represent visits to a Depot, collecting/delivering goods for one or more Jobs. Depot-Stops also represent a “run”.
- Job-Stops, representing Collection/Distribution of goods for a single Job at a customer-site or representing Periods of unavailability.
- Breaks, representing a lunch-Break etc.

Note: Not all statuses apply to all types, and their interpretation may vary:

LOGON/LOGOFF Stop Statuses

Token	Description
DESPATCHED	Not yet logged on to Shift or logged off from Shift
COMPLETED	Logged on to Shift or logged off from Shift.

The status for LOGON/LOGOFF Stops is wholly determined by the status of the associated SHIFT. If the SHIFT has never been logged on to, both Stops are DESPATCHED. If the SHIFT is currently logged on to, the LOGON Stop has status COMPLETED, and the LOGOF Stop DESPATCHED. If the SHIFT has been logged off, then both are COMPLETED.

DIST/COLL Depot Stop Statuses (Sheet 1 of 2)

Token	Description
ASSIGNED	Incorporated into the schedule, but not yet “allocated”. Scheduler can remove/replace the Stop at will.
ALLOCATED	Incorporated into the schedule, the Stop cannot be removed/replaced. However Scheduler is not required to maintain the run in its current form.
CLOSED	As ALLOCATED, but Scheduler may not make any changes to the composition of the run.

DIST/COLL Depot Stop Statuses (Continued) (Sheet 2 of 2)

ENROUTE	As CLOSED, but the Resource will visit this Stop next.
ARRIVED	As ENROUTE, the Resource has arrived at the site.
STARTED	As CLOSED, and the driver has commenced loading/unloading.
COMPLETED	As CLOSED, and the driver has completed loading/unloading.

PKUP/DROP Stop Statuses

Token	Description
INACTIVE	The Stop is defined as per normal, but not included into the schedule. Mainly used to define Stops for subsequent “Choose” operations.
FREE	Not assigned to any Shift, except for BREAK and LOGON,LOGOFF.
ASSIGNED	Incorporated into the schedule, but not yet “allocated”. Scheduler can move the Stop at will.
ALLOCATED	Incorporated into the schedule and attached to a SHIFT, Scheduler cannot change the SHIFT. However Scheduler is not required to keep the Stops associated with the same Depot-Stop.
CLOSED	As ALLOCATED, but Scheduler cannot change the associated Depot-Stop.
DESPATCHED	As CLOSED, and the driver has been advised.
ACKNOWLEDGED	As DESPATCHED, and the driver has acknowledged the Job.
ENROUTE	As CLOSED, and the Resource will visit this Stop next.
ARRIVED	As ENROUTE, but the Resource has arrived at the site.
POSTPONED	As CLOSED, this STOP cannot be started until the specified time.
STARTED	As CLOSED, and the driver has commenced loading/unloading.
COMPLETED	As CLOSED, and the driver has completed loading/unloading.

POU/PTP Stop Statuses (Sheet 1 of 2)

Token	Description
INACTIVE	The Stop is defined as per normal, but not included into the schedule. Mainly used to define Stops for subsequent “Choose” operations.
FREE	Not assigned to any Shift, except for BREAK and LOGON,LOGOFF.
ASSIGNED	Incorporated into the schedule, but not yet “allocated”. Scheduler can move the Stop at will.
ALLOCATED	Incorporated into the schedule and attached to a SHIFT, Scheduler cannot change the SHIFT.
DESPATCHED	As ALLOCATED, and the driver has been advised.
ACKNOWLEDGED	As DESPATCHED, and the driver has acknowledged the Job.

POU/PTP Stop Statuses (Continued) (Sheet 2 of 2)

ENROUTE	As ALLOCATED, but the Resource will visit this Stop next.
ARRIVED	As ALLOCATED, the Resource has arrived at the site.
POSTPONED	As ALLOCATED, this STOP cannot be started until the specified time.
STARTED	As ALLOCATED, and the driver has commenced loading/unloading.
COMPLETED	As ALLOCATED, and the driver has completed loading/unloading.

Warning: POU Stops are DISABLED the moment their status is changed to anything lesser than ALLOCATED.

BREAK Stop Statuses

Token	Description
FREE	The Break is not scheduled to be taken, i.e. Scheduler has not included the Break into the schedule
ALLOCATED	Scheduler has included the Break into the schedule.
STARTED	The driver has started, but not yet completed his Break
COMPLETED	The driver has completed his Break.

Group Status

Token	Description
INDEPENDENT	The Job is not associated with any other Jobs.
GROUPED	The Stop is associated with other Jobs. All Jobs with the same GROUP_ID must be kept within the same run., but non-member Stops may be mixed in. **Note:** note: conceptually only JOBS can be grouped, but as Scheduler only receives Stops, grouping info must be supplied in all Stops of a Job.
LINKED	The Job belongs to group GROUP. All Stops of the Group are adjacent. If the Stop-type is PKUP or COLL, the Group forms a run, i.e. no “non-member” Stops may be included in between the group and adjacent Depots.

7.12 Status Messages Packets

This section describes the Oracle Real-Time Scheduler message types and the packets for communicating the messages.

Message types

Token	Description
WARNING	Reports inconsistencies in incoming packets, or warns about unexpected situations during the scheduling process.
ERROR	Reports inconsistencies and worse situations in incoming packets. Aborts requested operation.
COMPLETED	Reports completion of the requested operation.
SYSTEM_WARNING	Switch sends Warnings to Users with SYSTEM level access in the event of a Primary Switch crash. SmAuto sends Warnings to Users with SYSTEM level access regarding expiry of Product Licence.
SYSTEM_MESSAGE	Switch sends Messages to Users with SYSTEM level access whenever a Switch process connects to it or disconnects from it.

The ‘ID” fields may be used to identify (and possibly select/retrieve) relevant objects. An ERROR-message indicates that an operation was aborted.

Packets communicating messages of type (WARNING and ERROR)

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Returns schedule dependent info.	WARNING, ERROR, COMPLETED
ORIGIN		STRING	Identifies the Client that sent the message	[]
DESTINATION		STRING	Identifies the Client the message must be sent to.	[broadcast to all]
PACKET		TOKEN	Identifies packet that caused the problem, if any.	
CODE		TOKEN	Error-Code.	
STOP_ID		IDENT	Stop involved, if any.	
RSRC_ID		IDENT	Resource involved, if any.	

Packets communicating messages of type (WARNING and ERROR)

Field Name	M	Format	Description	Range
SHIFT_ID		IDENT	Shift involved, if any.	
DEPOT_ID		IDENT	Depot involved, if any.	
BREAK_ID		IDENT	Break involved, if any.	
SLOT_ID		IDENT	Slot involved, if any.	
TEXT	X	STRING	Message in plain text.	

Packets communicating System Messages/Warning

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	System reports sent only to Users with SYSTEM level Access.	SYSTEM_WARNING SYSTEM_MESSAGE,
TEXT		STRING	Message in plain text.	

7.13 MANUAL PLANNING PACKETS

Manual planning packets assist and enable the Allocator to generate a plan or to provide real-time information (on arrival, departure etc.) to Scheduler.

Note: Most operations below operate on Groups. However, since an “ungrouped” Job forms a Group in its own right, these operations also act on single, ungrouped, Jobs. As Oracle Real-Time Scheduler does not recognize levels of grouping a single “UNGROUP” operation will dissolve all levels of Groups at once.

Packets facilitating Manual Planning (Sheet 1 of 4)

Packet Name	Requirement	Requests SmAuto to:
RSRC_POSITION	PLAN-REALTIME-RESOURCE-POSITION	Set a new position for a Resource.
RSRC_DELAY	PLAN-REALTIME-RESOURCE-DELAY	Delay the progress of a Resource.
RSRC_DELAY_COMPLETE		Make the delayed Resource available again.
SHIFT_DISABLE	PLAN-SHIFT-ENABLE	Disable a Shift (status=INACTIVE). The Shift can no longer be used for planning purposes.
SHIFT_ENABLE		Set the Shift-status to PLANNED.
SHIFT_START	PLAN-REALTIME-SHIFT-START	Set the Shift-status to STARTED to COMPLETING, depending on whether the Shift has been CLOSED.
SHIFT_COMPLETE	PLAN-REALTIME-SHIFT-COMPLETE	Set the Shift-status to COMPLETED. No Jobs that are not COMPLETED may be associated with the Shift.
SHIFT_CLOSE	PLAN-SHIFT-CLOSE	Set the Shift-status from PLANNED to CLOSED, or from STARTED to COMPLETING.
SHIFT_OPEN		Reverse a previous “SHIFT_CLOSE”.
SHIFT_OPTIMIZE	PLAN-SHIFT-OPTIMIZE	Optimize the order all Stops assigned to a Shift. If the Shift is part of a “Haul”, all of the Haul will be optimized.
SHIFT_STANDBY		Set the Shift-status to STANDBY. The Shift will not be used for planning purposes until logged on.
BREAK_DISABLE	PLAN-BREAK-DISABLE	Disable a Break (status=INACTIVE). The Break will not be considered for planning-purposes.
BREAK_ENABLE		Set the Break-status to FREE.

Packets facilitating Manual Planning (Continued) (Sheet 2 of 4)

BREAK_START		Set the Break-status to STARTED.
BREAK_COMPLETE		Set the Break-status to COMPLETED.
RUN_CLOSE	PLAN_RUN_CLOSE	Close a run, i.e. only Jobs that are at least ALLOCATED can be part of this run, preventing Scheduler from assigning new Stops to the run.
DEPOT_CUTOFF		Closes runs starting before CutOff
DEPOT_CUTOFF_CONFIRMED		Confirming assignment of a Job.
GROUP	PLAN-JOB-GROUP	Group Jobs and/or groups into a single group.
UNGROUP	PLAN-JOB-UNGROUP	Ungroup all Jobs of a group.
UNGROUP_JOBS	PLAN-JOB-UNGROUP-JOBS	Remove one or more Jobs from a group.
JOB_DISABLE	PLAN-JOB-DISABLE	Disable a Job (status==INACTIVE). The Job cannot be scheduled.
JOB_FREE	PLAN-JOB-FREE	Set the Job-status to FREE. Scheduler can assign the Job to any compatible Shift.
JOB_ASSIGN	PLAN-JOB-ASSIGN	Associate the Job to a specified Shift. However, Scheduler can change the assignment at will.
JOB_ALLOCATE	PLAN-JOB-ALLOCATE	Associate the Job to a specified Shift. In this case Scheduler **cannot** change the association.
JOB_CLOSE	PLAN-JOB-FIX-RUN	Associate the Job with a single run (Depot-Stop) and Shift. Scheduler cannot change the associated Shift or Depot-Stop.
JOB_ASSIGN_COND		Insert the Job in the schedule if the cost of doing so is below MAX_COST.
CA_JOB_ASSIGN_COND		Insert the Job in the schedule if the cost of doing so is below MAX_COST.
JOB_TRANSFER		Transfers Job to another Depot
JOB_DESPATCH	PLAN-REALTIME-JOB-DESPATCH	Associate the Job with a single run (Depot-Stop) and Shift. In addition to JOB_CLOSE, the driver is notified.
JOB_DESPATCH_RUN		As JOB_DESPATCH, but despatches all Jobs associated with the run.
JOB_DRIVER_ACK		Set the Job-status to ACKNOWLEDGED, indicating that the driver is aware of the association.

Packets facilitating Manual Planning (Continued) (Sheet 3 of 4)

JOB_LOAD	PLAN_JOB_LOAD	Updates the Loads specified for Jobs
STOP_DESPATCH		Undo an earlier ENROUTE, ARRIVE, POSTPONE, START or COMPLETE.
STOP_ENROUTE	PLAN-REALTIME-RESOURCE-ENROUTE	Place the Stop first in the driver’s itinerary.
STOP_ETA_UPDATE		Supply external ETA for Stop into SmAuto from plannerClient.
STOP_ARRIVE		Set the Stop to ARRIVED.
STOP_POSTPONE		postpone the Stop until a specified time.
STOP_START		Set the Stop to STARTED.
STOP_COMPLETE	PLAN-REALTIME-STOP-COMPLETE	Set the Stop to COMPLETED.
JOB_FIX_HAUL	PLAN-JOB-FIX-HAUL	Restrict the Job to being associated with the specified Haul.
JOB_FIX_RSRC	PLAN-JOB-FIX-RSRC	Restrict the Job to being associated with the specified Resource.
JOB_LINK	PLAN-JOB-LINK	Keep all Stops of a Job/group together.
JOB_UNLINK	PLAN_JOB_UNLINK	Unlinks the Stops of a Job/group.
JOB_UNSEQ	PLAN_JOB_UNSEQ	Unfreezes the order of Stops in a Job/group.
JOB_SEQUENCE	PLAN-JOB-SEQUENCE	To freeze the order of Stops belonging to the a Job/group.
JOB_INSERT	PLAN-JOB-INSERT	insert a Job at a specified location.
STOP_INSERT	PLAN-STOP-INSERT	Move a Stop to a specified location within the Haul.
CHS_SHIFT		Returns a list of possible single Stop time slots in the schedule, but only one per Shift.
CHS_SLOT		Find and return time slots.
CHS_REPLY		Chooser reply-packet containing a single time slot.
CHS_REPLY_END		There will be no further reply packets.

Packets facilitating Manual Planning (Continued) (Sheet 4 of 4)

GO_HOME		Instructs SmAuto that driver can go home once there is no more work for him by setting the Shift's status to "Going Home".
RESUME_WORK		Instructs SmAuto to cancel the go home instruction sent earlier to driver.
BIND_STOP		Create a Bound Job
UNBIND_STOP		Remove the Stop from the Bound Job.
SET_SUBWINDOW		Specify Subwindow width for Bound Jobs.
TIGHTEN_BOND		Perform automatic tightening on Bound Jobs.

7.13.1 Manual Resource operations

The following two packets facilitate manual repositioning and delaying of Resources.

Manually (Reposition) Resource

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Places a Resource at a new position and time.	RSRC_POSITION
IDENT	X	IDENT	Identifies the Resource.	
LATITUDE	X	DOUBLE	Position of the Resource. Accurate positioning requires 6 significant digits after the decimal.	
LONGITUDE				
DATE		DATE	Date and time of day at which the Resource was at the specified position. Either both or none must be present.	yyyymmdd hhmm [NOW] If time zones are used then the range will be: (hhmmZ-hhmm) to (hhmmZ+hhmm)
TIME		HHMM		

Manually (Delay) Resource

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Delays a Resource until a defined Date and Time.	RSRC_DELAY
IDENT	X	IDENT	Identifies the Resource.	
DATE	X	DATE	Date and time of day until which the Resource will be delayed.	yyyymmdd hhmm If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)). That is: ((local time - UTC offset) to (local time + UTC offset))
TIME		HHMM		

Manually (Un-Delay) Resource

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Makes a delayed Resource available for work once again.	RSRC_DELAY_COMPLETE
IDENT	X	IDENT	Identifies the Resource.	

7.13.2 Manual Shift operations

Packets in this section enable the Allocator to change the status of a Shift manually.

Manually (Start/Complete) Shift

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Set the Shift-status to STARTED to COMPLETING, depending on whether the Shift has been CLOSED.	SHIFT_START
			Set the Shift-status to COMPLETED. No Jobs that are not COMPLETED may be associated with the Shift	SHIFT_COMPLETE
SHIFT	X	IDENT	Identifies the Shift operated on	
DATE		DATE	Date and time of the event. If not specified, NOW will be substituted when ***advancing*** the status and the previous value when ***reversing***.	yyyymmdd hhmm [previous or NOW] If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)). That is: ((local time - UTC offset) to (local time + UTC offset))
TIME		HHMM		

Manually change Shift status to (Disable/Enable/Close/Open/Optimize/Standby)

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Disable a Shift (status=INACTIVE). The Shift can no longer be used for planning purposes.	SHIFT_DISABLE
			Set the Shift-status to PLANNED.	SHIFT_ENABLE
			Set he Shift-status from PLANNED to CLOSED, or from STARTED to COMPLETING.	SHIFT_CLOSE
			Set he Shift-status from PLANNED to CLOSED, or from STARTED to COMPLETING.	SHIFT_OPEN
			Optimizes the order of all Stops on a Shift. If the Shift is part of a “Haul”, all of the Haul will be optimized.	SHIFT_OPTIMIZE
			Set the Shift-status to STANDBY. The Shift will not be used for planning purposes until logged on.	SHIFT_STANDBY
SHIFT	X	IDENT	Identifies the Shift operated on.	

7.13.3 Manual Break operations

Packets in this section enable the Allocator to change the status of a Break manually.

Manually (Disable/Enable) Break

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Disable a Break (status=INACTIVE). The Break can no longer be used for planning purposes.	BREAK_DISABLE
			Set the Break-status to FREE.	BREAK_ENABLE
BREAK	X	IDENT		

Manually (Start) Break

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Set the Break-status to STARTED.	BREAK_START
BREAK	X	IDENT		
DATE		DATE	Date and time of day at which the Break was started. If the Break-status is COMPLETED, this field defaults to the previous start-time.	yyyymmdd hhmm [previous or NOW] If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)). That is: ((local time - UTC offset) to (local time + UTC offset))
TIME		HHMM		
LATITUDE		DOUBLE	Position where Break was taken.	[previous or current]
LONGITUDE				

Manually (Complete) Break

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Set the Break-status to COMPLETED.	BREAK_COMPLETE
BREAK	X	IDENT		

Manually (Complete) Break

Field Name	M	Format	Description	Range
COMPL_DATE		DATE	Date and time of day at which the Break was completed.	yyyymmdd hhmm [NOW] If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)). That is: ((local time - UTC offset) to (local time + UTC offset))
COMPL_TIME		HHMM		
START_DATE		DATE	Date and time of day at which the Break was started. If the Break-status is STARTED, this field defaults to the previous start-time.	yyyymmdd hhmm [previous] If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)). That is: ((local time - UTC offset) to (local time + UTC offset))
START_TIME		HHMM		
LATITUDE		DOUBLE	Position where Break was taken.	[previous or current]
LONGITUDE				

7.13.4 Manual Run operations

There is only manual operation possible on a run is to “close” it.

Manually (Close) Run

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Close a run. Only manual operations can be performed on a closed run. Scheduler cannot change the set of Stops associated with the run.	RUN_CLOSE
DEPOT_STOP	X	IDENT	Ident of the Depot-Stop representing the run.	

7.13.5 Manually Close Multiple Runs

The DEPOT_CUTOFF packet will trigger Oracle Real-Time Scheduler to close all runs that currently have a departure time earlier than the time specified in the CUTOFF_DATE/ TIME field.

Manually Specify Depot Cut-Off

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Closes runs starting before CutOff.	DEPOT_CUTOFF
IDENT		IDENT	Uniquely identifies the Depot to which the CutOff applies. If omitted, CutOff will be applied to all Depots.	
CUTOFF_DATE	X	DATE	All runs starting before this time of the day, will be closed.	yyyymmdd
CUTOFF_TIME	X	HHMM		hhmm If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)). That is: ((local time - UTC offset) to (local time + UTC offset))

Depot Cutoff Confirmation from Scheduler

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Confirming assignment of a Job.	DEPOT_CUTOFF_CONFIRMED
IDENT		IDENT	Identifies the Depot.	

7.13.6 Manual Group operations

“Group” is the term used to indicate a set of Jobs which for all intend and purpose acts as a single Job, i.e. manual operations act on all Jobs in the group, all Jobs must be allocated to the same Haul and the same Depot-Stop.

Manually (Group) Jobs

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	A request for SmAuto to group Jobs.	GROUP
JOBS	X	IDENT	List of Job-id’s to be grouped. If any of the Jobs already belong to a Group, all Jobs belonging to this Group will be incorporated into the new Group.	

Manually (UnGroup all) Jobs in Group

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Removes all Jobs from the Group <JOB> belongs to.	UNGROUP
JOB	X	IDENT	The id of a member of the Group.	

Manually (UnGroup Specified) Jobs in Group

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Removes the specified Jobs only from a group.	UNGROUP_JOBS
JOBS	X	IDENT	The id's of the Jobs to be removed.	

7.13.7 Manual Job operations

Packets in this section enable you to change the Job status manually.

Manually (Disable/Free/Driver_Acknowledge) Job

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Disable a Job (status==INACTIVE). The Job cannot be scheduled.	JOB_DISABLE
			Set the Job-status to FREE. Scheduler can assign the Job to any compatible Shift.	JOB_FREE
			Set the Job-status to ACKNOWLEDGED, indicating that the driver is aware of an earlier DESPATCH.	JOB_DRIVER_ACK
JOB	X	IDENT		

Manually (Assign/Allocate) Job

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Undo an earlier ALLOCATE, CLOSE etc. No change to the current assignments, but Scheduler can change the assignment at will.	JOB_ASSIGN
			The current association of the Stops of the specified Job/group can no longer be changed by Scheduler.	JOB_ALLOCATE
JOB	X	IDENT	Job to be operated on.	
SHIFT		IDENT	Shift to which the Job is to be assigned/allocated. If no Shift, or the Shift the Job is currently on is specified, just the Job’s status will be changed. Otherwise the Job will be transferred to the new Shift, and the Shift will be re-optimized.	[current]

7.13.7.1 Conditional Assignment of Job

While the JOB_ASSIGN packet represents an unconditional insertion into the schedule, the JOB_ASSIGN_COND and CA_JOB_ASSIGN_COND packets instruct the Scheduler to insert the Job into the schedule if the cost of doing so does not exceed a specified amount (MAXCOST), and leave it INACTIVE otherwise.

The INITIAL_STATUS field in the STOP packet (see “INITIAL_STATUS” on page 7 - 61) dissuades Scheduler from assigning the Job to a Shift while the Job is in Scheduler (after receipt of BATCH_OFF). The Job must be assigned as a result of the JOB_ASSIGN_COND or CA_JOB_ASSIGN_COND packets only.

7.13.7.1.1 JOB_ASSIGN_COND

JOB_ASSIGN_COND request packet maintains backward compatibility with Planner versions 10.4 and below and RM version 4.4 and below.

To make a Conditional Assignment Request using JOB_ASSIGN_COND packet the following sequence of packets must be sent:

- BATCH_ON
- STOP
- STOP_TW
- BATCH_OFF
- JOB_ASSIGN_COND

The deletion of the Task in case of failure is controlled by a command line argument "deleteRejectedJob". Please refer to the accompanying Planner System Manual for more details on Planner command line arguments.

Conditional Assignment of Job

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Insert the Job in the schedule if the cost of doing so is below MAX_COST.	JOB_ASSIGN_COND
JOB	X	IDENT	Job to be operated on.	
SHIFT	X	IDENT	Shift to which the Job is to be assigned. The Job will be transferred to the new Shift, which will be re-optimized.	
MAXCOST	X	DOUBLE	The cost below which the assignment will take place.	>0, 1e-10
FIND_SHIFT		TOKEN	If the Job cannot be assigned to the Shift below MAX_COST, but it is possible to find another Shift within the constraint, assign to that Shift instead.	[NO],YES

7.13.7.1.2 CA_JOB_ASSIGN_COND

To make a Conditional Assignment Request using CA_JOB_ASSIGN_COND packet the following sequence of packets must be sent:

- CA_BATCH_ON
- STOP
- STOP_TW
- ...Multiple sets of STOP and STOP_TW packets can be included here
- CA_JOB_ASSIGN_COND
- CA_BATCH_OFF

Multiple STOP packets belonging to the same Job can be included within the CA_BATCH_ON and CA_BATCH_OFF sentinel element.

Note: When a Multi-STOP Job is included within a single CA_BATCH_ON and CA_BATCH_OFF sentinal element, the reply is processed for the STOP specified within the CA_JOB_ASSIGN_COND packet

As a result of the Conditional Assignment Request using CA_JOB_ASSIGN_COND packet if a new task is rejected, then the deletion of this task will be controlled by the "deleteRejectedJob" argument.

Conditional Assignment of Job

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Insert the Job in the schedule if the cost of doing so is below MAX_COST.	CA_JOB_ASSIGN_COND
JOB	X	IDENT	Job to be operated on.	
SHIFT	X	IDENT	Shift to which the Job is to be assigned. The Job will be transferred to the new Shift, which will be re-optimized.	
MAXCOST	X	DOUBLE	The cost below which the assignment will take place.	>0, 1e-10
FIND_SHIFT		TOKEN	If the Job cannot be assigned to the Shift below MAX_COST, but it is possible to find another Shift within the constraint, assign to that Shift instead.	[NO],YES

7.13.7.2 Handshake packets

The Switch responds to the originators of JOB_ASSIGN_COND request, JOB_ALLOC_COND request and CA_JOB_ALLOC_COND request using the JOB_ACCEPTED, JOB_REJECTED or JOB_CONFIRMED packets.

JOB_ACCEPTED/REJECTED/CONFIRMED

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Confirming assignment or allocation of Job	JOB_ACCEPTED
			Confirming rejection of Job	JOB_REJECTED
			Confirming that an accepted Job has been transferred to the secondary Switch	JOB_CONFIRMED
JOB_ID	X	IDENT	Ident of Job that is being confirmed	

7.13.7.3 Conditional Allocation of Job

The rules for Conditionally Allocating Jobs within Oracle Real-Time Scheduler is described in this section.

1Oracle Real-Time Scheduler receives a request to allocate a number of Jobs to a Shift.

2If the allocation would violate any hard constraints, then the request is rejected and an error is returned. For example: The Shift does not have refrigerator but the Job requires refrigeration.

3If the allocation would exceed a preset cost (excessive violation of soft constraints), then the request is rejected and an error is returned. For example: Allocating the Job to the shift will cause the overtime cost to exceed the preset cost (MAXCOST).

4Otherwise Oracle Real-Time Scheduler will allocate the Jobs to the specified Shift and return success.

5In case of one or more hard constraint error, Oracle Real-Time Scheduler will return an error detailing all violations for each Job.

6In case of a soft constraint violation, Oracle Real-Time Scheduler will return as much detail as possible, but will not identify individual Jobs.

As Oracle Real-Time Scheduler prevents all constraint violations associated with the conditional allocation of Jobs, the user can select and fix a Shift for existing Jobs, without having to validate the compatibility of the selected Shift.

Conditional Allocation of Job

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Fix the Jobs to the selected Shift in the schedule if the cost of doing so is below MAX_COST.	JOB_ALLOC_COND
JOBS	X	IDENT	Jobs to be operated on.	
SHIFT	X	IDENT	Shift to which the Jobs are to be allocated. The Jobs will be transferred to the new Shift, and the Shift will be re-optimized.	
MAXCOST	X	DOUBLE	The cost below which the allocation will take place.	>0

7.13.7.4 Manually Alter Jobs to Depots Association

The JOB_TRANSFER packet instructs the Scheduler to transfer the Job specified in the JOB field to the Depot or list of Depots specified in the DEPOT field.

In other words the JOB_TRANSFER packet changes the Depot or list of Depots associated with the Job.

Manually Altering Depot Job Association

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Transfers Job to another Depot	JOB_TRANSFER
JOB	X	IDENT	Identifies the Job to be transferred.	
DEPOT	X	IDENT	Comma separated list of Depots to which the Job can be transferred.	

7.13.7.5 Manually Close and Despatch Jobs

Manually (Close) Job

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	A request for SmAuto to group Jobs.	JOB_CLOSE
JOB	X	IDENT	As JOB_ALLOCATE, but additionally the Job is associated with a Depot-Stop.	
DEPOT_STOP		IDENT	Ident of the Depot-Stop to which the Job is to be closed.	[current]

Manually (Despatch) Job

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Same as in JOB_CLOSE, but additionally the driver is notified.	JOB_DESPATCH
JOB	X	IDENT	Job to be despatched.	
SHIFT		IDENT	Shift to which the Job is to be despatched. If no Shift, or the Shift the Job is currently on is specified, just the Job's status will be changed. Otherwise the Job will be transferred to the new Shift, and the Shift will be re-optimized.	[current]

Manually (Despatch) Job

Field Name	M	Format	Description	Range
TEXT		STRING	A message to be send to the driver as part of the “despatch-message”. The “despatch-message” will be send when the Job-status is ***advanced*** to DESPATCHED, or when the Job is despatched to a Shift other than “current”.	

Manually (Despatch) Jobs in Run

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	As JOB_DESPATCH, but despatches all Jobs associated with the run.	JOB_DESPATCH_RUN
RUN_ID	X	IDENT	Identifies the run that is being despatched. (IDENT of Depot-Stop)	
TEXT		STRING	Text to be sent with the despatch message(s).	

7.13.7.6 Manually Updating Job’s Load

Oracle Real-Time Scheduler can limit the number of deliveries from a Depot on any day to the delivery capacity set for that Depot and day. See Depot capacity under “Depot Service TW” on page 7 - 59.

Using the JOB_LOAD packet Oracle Real-Time Scheduler limits the number of Items delivered from a Depot on any day to the picking capacity set for that Depot and that day.

Manually (Updating Job’s Load)

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Updates the load attributes of a Job. The Job may be single-Stop distribution or collection Job, or a 2-Stop Point-To-Point.	JOB_LOAD
IDENT	X	IDENT	Job Ident.	
WEIGHT		DOUBLE	New weight for the Job. The sign (+/-) of this value is ignored.	[as is]
VOLUME		DOUBLE	New volume for the Job. The sign (+/-) of this value is ignored.	[as is]
SIZE		DOUBLE	New size for the Job.	>=0 [as is]

7.13.7.7 Manually Restrict Job Assignment

Packets under this section restrict the associations between a Job and other objects.

Manually Fix/Unfix a Job-Haul Association

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Restrict the Job to being assigned to the specified Haul only	JOB_FIX_HAUL
JOB	X	IDENT		
HAUL		IDENT	Omitting the Haul-ident cancels an earlier restriction.	

Manually Fix/Unfix a Job-Resource Association

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Restrict the Job to being assigned to the specified Resource only	JOB_FIX_RSRC
JOB	X	IDENT		
RSRC		IDENT	Omitting the Resource-ident cancels an earlier restriction.	

7.13.8 Manual Stop operations

Packets in this section enable manual operations on Stops.

Manually (Despatch) Stop

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Undo an earlier ENROUTE, ARRIVE, POSTPONE, START or COMPLETE. by setting the status to DRIVER_ACK.	STOP_DESPATCH
STOP	X	IDENT	Identifies the Stop	

Manually (Enroute) Stop

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Place the Stop first in the driver’s itinerary.	STOP_ENROUTE
STOP	X	IDENT	Identifies the Stop	

Manually (Update ETA for) Stop

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Updates the Stop’s "external" ETA into SmAuto from the plannerClient	STOP_ETA_UPDATE
STOP	X	IDENT	Identifies the Stop	
ETA_DATE	X	DATE	Specifies the ETA date	
ETA_TIME	X	HHMMSS	Specifies the ETA time of the day	If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)). That is: ((local time - UTC offset) to (local time + UTC offset))
spare		STRING	May be used to extend STOP_ETA_UPDATE packet without causing incompatibility with previous versions	
spare		STRING		
spare		STRING		

Manually (Arrive) Stop

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	The Resource has arrived at the Stop.	STOP_ARRIVE
STOP	X	IDENT		
DATE		DATE	Date and time of the day at which the Resource arrived at the Stop. If not specified, and this message “undoes” one of the subsequent statuses, the stored arrival-time is used, otherwise NOW.	yyyymmdd hhmm [previous or NOW] If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)). That is: ((local time - UTC offset) to (local time + UTC offset))
TIME		HHMM		

Manually (Postpone) Stop

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	The Stop cannot be visited until the specified time.	STOP_POSTPONE
STOP	X	IDENT		
DATE	X	DATE	Date and time of the day for the earliest arrival at the Stop. **note**: *The actual time-window of the Stop will not change. Therefore, if the Stop is postponed beyond the specified time-window, the Stop will always be late.*	yyyymmdd hhmm
TIME		HHMM		

Manually (Start) Stop

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	A Stop has been started.	STOP_START
STOP	X	IDENT		
DATE		DATE	Date and time of day at which the Stop was started. If not specified, and this message “undoes” one of the subsequent statuses, the stored start-time is used, otherwise NOW.	yyyymmdd hhmm [previous or NOW]
TIME		HHMM		

Manually (Complete) Stop

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Set the Stop-status to COMPLETED.	STOP_COMPLETE

Manually (Complete) Stop

Field Name	M	Format	Description	Range
STOP	X	IDENT		
COMPL_DATE		DATE	Date and time of the day at which the Stop was completed.	yyyymmdd hhmm [NOW]
COMPL_TIME		HHMM		
START_DATE		DATE	Date and time of the day at which the Stop was started. If the Stop-status is STARTED, this field defaults to the previous start-time.	yyyymmdd hhmm [previous] If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)). That is: ((local time - UTC offset) to (local time + UTC offset))
START_TIME		HHMM		

7.13.8.1 Manually Restrict the Ordering of Stops in Schedule

Packets in this section enables the Allocator to manually restrict the ordering of Stops in a Group.

Manually (Link/UnLink, Sequence/UnSequence) Jobs

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Scheduler can no longer insert “foreign” Stops within the group.	JOB_LINK
			Scheduler can no longer change the order of Stops within the group.	JOB_SEQUENCE
			Undo LINK	JOB_UNLINK
			Undo SEQUENCE	JOB_UNSEQ
JOB	X	IDENT	The Job to be operated on. If the Job is part of a group, the operation will be applied to the whole group	

7.13.8.2 Manually Inserting Stops into Plan

Packets in this section enable Stops to be manually inserted at a specific position in the schedule.

Manually Insert Stops within any Haul

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Inserts a Stop immediately prior to the named Stop. This packet inserts Stops within any Haul.	JOB_INSERT
JOB	X	IDENT	Ident of the Job to be inserted.	
TARGET	X	IDENT	Ident of the Stop to insert immediately before.	

Manually Insert Stops in the same Haul

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Inserts a Stop immediately prior to the named Stop. This packet inserts Stops only within the same Haul.	STOP_INSERT
STOP	X	IDENT	Ident of the Stop to be inserted.	
TARGET	X	IDENT	Ident of the Stop to insert immediately before.	

7.13.9 Chooser Operations for Manual Planning

The CHS_???? and SR_CHS_SLOT packets calculate and return a list of possible inclusions for a single Stop in order of effectiveness. The requested Stop can have any status other than the COMPLETED status.

Note: These packets leave the actual schedule unchanged.

CHS_SHIFT Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Find and return inclusion points, limited to one per Shift.	CHS_SHIFT
IDENT	X	IDENT	Identifies the requesting packet. This id will be returned with each returned inclusion-point.	
STOP	X	IDENT	IDENT of the Stop to insert.	
MAX_COUNT		CARDINAL	The maximum number of options to return.	>=1; [1]
MAX_UNITS		DOUBLE	The maximum number cost differential between the first and the last inclusion points that were returned.	>0; [HUGE]
spare			May be used to extend CHS_SHIFT packet without causing incompatibility with previous versions	
spare				
spare				

7.13.9.1 Chooser Slot Request

The CHS_SLOT nad SR_CHS_SLOT packets request Oracle Real-Time Scheduler to respond with a set of distinct delivery slots that would create the lowest cost when inserted into the Schedule.

7.13.9.1.1 CHS_SLOT

CHS_SLOT request packets maintain backward compatibility with Planner versions 10.4 and below and RM version 4.4 and below.

To make an Appointment Booking Request using CHS_SLOT packet the following sequence of packets must be sent:

- BATCH_ON
- STOP
- STOP_TW
- BATCH_OFF
- CHS_SLOT

This is usually followed by a delete STOP packet sequence as listed below:

- BATCH_ON
- STOP_DEL (this is the same STOP as in the Slot Request sequence above)
- BATCH_OFF

CHS_SLOT Request Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Finds and returns time slots for stops.	CHS_SLOT
IDENT	X	IDENT	Identifies the requesting packet. This id will be returned with each returned time slot.	
STOP	X	IDENT	IDENT of the Stop to insert.	
SLOT_GROUP	X	IDENT	Identifies the group of slots applicable to this Stop.	
MAX_COUNT		CARDINAL	The maximum number of time slots to return.	>=1; [1]
MAX_UNITS		DOUBLE	After the first time slot, Oracle Real-Time Scheduler will not return any options with a greater cost.	>0.1; [HUGE]
spare		STRING	May be used to extend CHS_SLOT packet without causing incompatibility with previous versions	
spare				
spare				

7.13.9.1.2 SR_CHS_SLOT

To make an Appointment Booking Request using SR_CHS_SLOT packet the following sequence of packets must be sent:

- SR_BATCH_ON
- STOP
- STOP_TW
- STOP_TW
- ...Multiple sets of STOP and STOP_TW packets can be included here
- SR_CHS_SLOT
- SR_BATCH_OFF

Multiple STOP packets belonging to the same Job can be included within the SR_BATCH_ON and SR_BATCH_OFF sentinel element.

Note: When a Multi-STOP Job is included within a single SR_BATCH_ON and SR_BATCH_OFF sentinal element, the reply is processed for the STOP specified within the SR_CHS_SLOT packet.

SR_CHS_SLOT Request Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Finds and returns time slots for stops.	SR_CHS_SLOT
IDENT	X	IDENT	Identifies the requesting packet. This id will be returned with each returned time slot.	
STOP	X	IDENT	IDENT of the Stop to insert.	
SLOT_GROUP	X	IDENT	Identifies the group of slots applicable to this Stop.	
MAX_COUNT		CARDINAL	The maximum number of time slots to return.	>=1; [1]
MAX_UNITS		DOUBLE	After the first time slot, Oracle Real-Time Scheduler will not return any options with a greater cost.	>0.1; [1e99]
spare		STRING	May be used to extend SR_CHS_SLOT packet without causing incompatibility with previous versions	
spare				
spare				

7.13.9.2 Chooser Reply Packet

Chooser reply packets report on the result of a “Choose” operation. Each reply contains a single proposed time slot within the schedule. Multiple time slots will be received in rank order, the best result first

CHS_RELPY Packet (Sheet 1 of 2)

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Returns a single time slot.	CHS_REPLY
IDENT	X	IDENT	Identifies the requesting packet.	
CHS_TYPE	X	TOKEN	Type of request generating this response.	CHS_SHIFT, CHS_SLOT, SR_CHS_SLOT
STOP	X	IDENT	IDENT of the Stop for which the request was performed.	
JOB		IDENT	IDENT of the related Job.	
SHIFT		IDENT	IDENT of Shift within which the first non-Depot Stop is visited.	
HAUL		IDENT	IDENT of Haul to which Stop inclusion is suggested.	
DEPOT		IDENT	IDENT of Depot at which goods for this Stop are loaded/dropped, or in case of a DEPOT-Stop, the associated Depot.	
DEPOT_STOP		IDENT	IDENT of Depot-Stop at which goods for this Stop are loaded/dropped.	
NEXTSTOP	X	IDENT	IDENT of Stop which is to immediately follow subject Stop when inserted.	
ARRDATE	X	DATE	ETA. Date and time of the day of expected arrival at the Stop.	yyyymmdd
ARRTIME		HHMMSS		hhmmss If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)). That is: ((local time - UTC offset) to (local time + UTC offset))

CHS_RELPY Packet (Continued) (Sheet 2 of 2)

Field Name	M	Format	Description	Range
DEPDATE	X	DATE	ETD. Date and time of the day of expected departure from the Stop.	yyyymmdd
DEPTIME		HHMMSS		hhmmss If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)). That is: ((local time - UTC offset) to (local time + UTC offset))
RANK	X	CARDINAL	Rank of this result.	>=1;
UNITS	X	DOUBLE	Approximate increase in the cost of the schedule when the Job is inserted as specified in this packet.	
NEW_RUN	X	TOKEN	Does Scheduler have to generate an additional run to accommodate this Stop?	YES/NO
NEW_SHIFT	X	TOKEN	Does Scheduler have to activate an additional Shift to accommodate this Stop?	YES/NO
SLOT_IDENT		IDENT	Identifies the selected slot by the identifier supplied when installing the slots.	
SLOT_DATE		DATE	Date, time of day and length of the Time-window associated with the selected slot.	yyyymmdd
SLOT_TIME		HHMM		hhmm If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
SLOT_DURATION		HHMM		hhmm
spare		STRING	May be used to extend CHS_REPLY packet without causing incompatibility with previous versions	
spare				
spare				

7.13.9.3 Chooser Reply Termination Packet

Oracle Real-Time Scheduler returns the result of a Slot request through the CHS_REPLY packet and terminates the sequence with a CHS_REPLY_END packet.

Chooser Reply Termination Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	There will be no further reply packets.	CHS_REPLY_END
IDENT	X	IDENT	Identifies the requesting packet.	
CHS_TYPE	X	TOKEN	Type of request generating this response.	CHS_SHIFT, CHS_SLOT, SR_CHS_SLOT

7.13.10 Instructing Shift to GO_HOME/RESUME_WORK

The "Go Home" command changes sets of Jobs, which has been assigned to the Shift. It instructs SmAuto that driver can go home once there is no more work for him by setting the Shift status to "Going Home". A "Go Home" message is sent to the driver if the driver is "Going Home", has MDT available and only once he finishes all jobs despatched to him. Please see 7.4.11 “Mobile Data Messages” on page 7 - 13 for related packets.

Note: Though not a necessary requirement, it is more realistic to use "Go Home" functionality when Dead Reckoning is ON and MDTs are available.

GO_HOME packet sent from Client

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Instructs driver to start moving to logoff. This packet will in time set the MUST_GO_HOME flag	GO_HOME
SHIFT	X	IDENT	Identifies the Shift driver is to go home on completion of all despatched work	
spare		STRING	Spare fields can be used in future, e.g. to identify next destination for longhaul shifts.	
spare				
spare				

RESUME_WORK packet sent from Client

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Instructs driver to stop moving to logoff.	RESUME_WORK
SHIFT	X	IDENT	Identifies the Shift driver is doing	
spare		STRING	Spare fields can be used in future.	
spare				
spare				

The "Resume Work" instruction will cancel the previous "Go Home" instruction.

7.13.11 Binding / UnBinding STOPs

On receiving the following packet SmAuto will check that the STOP (STOP_ID) is not currently a primary STOP for other STOPs and that the primary STOP (PRIMARY_STOP_ID) does not have some other primary STOP specified.

Bind Stop Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Create a Bound Job	BIND_STOP
STOP_ID	X	IDENT	Uniquely identifies stop.	
PRIMARY_STOP_ID	X	IDENT	Uniquely identifies primary stop for the order.	
MIN_OFFSET		HHMM	Minimum time period to offset from the Primary’s stop time window.	hhmm[0]
MAX_OFFSET		HHMM	Maximum time period to offset from the Primary’s stop time window. MAX_OFFSET >= MIN_OFFSET	hhmm[MIN_OFFSET]

On receiving the following packet SmAuto will check that the STOP is a secondary STOP.

Unbind Stop Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Remove the Stop from the Bound Job.	UNBIND_STOP
STOP_ID	X	IDENT	Uniquely identifies stop.	

7.13.11.1 Manual setting of SubWindow

On receiving the following packet, SmAuto will check that the STOP is a primary STOP. This packet will not initiate AutoTightening process is just sets SubWindow for a primary STOP (with subsequent automatic adjustment of all secondaries), but otherwise leave all the rest to the Optimiser.

Set SubWindow

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Specify the Subwindow width for the Bound Jobs.	SET_SUBWINDOW
STOP_ID	X	IDENT	Uniquely identifies stop.	
SUB_WIN_DATE	X	DATE	Defines a SubWindow date.	

Set SubWindow

Field Name	M	Format	Description	Range
SUB_WIN_TIME	X	HHMM	Defines the time of the day when the SubWindow will start.	hhmm If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
SUB_WIN_DURATION	X	HHMM	Defines the length of the SubWindow.	hhmm

7.13.11.2 Manual activation of AutoTightening process

On receiving the following packet SmAuto will check that the STOP is a primary STOP. This packet will result in the attempt to find new SubWindows (optimal for the current Plan) and assign the whole Order.

Activate BondTightening Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Perform Automatic tightening of Bound Jobs.	TIGHTEN_BOND
PRIMARY_STOP_ID	X	IDENT	Uniquely identifies primary stop.	

7.14 Mobile Data Terminal Text Message Packets

“Mobile Data” packets are used for communication between Oracle Real-Time Scheduler and the drivers.

7.14.1 Text messages

These messages are used to convey text-information between the Allocator and the driver. The bulk of these messages will consist of despatch-messages, but there is provision for free-format messages from Allocator to driver and vice-versa.

7.14.1.1 Client initiated message

Client Sends message to MDT

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Sends a message to an MDT.	REQ_TEXT_TO_ MDT
SHIFT_ID		IDENT	Identifies the destination of the “text” by ident of the receiving Shift. note: Ignored when MDT_ID supplied, required if not.	
MDT_ID		IDENT	Identifies the destination of the “text” by the ident of the receiving device.	
TEXT	X	STRING	Free format text to be displayed on the receiving device.	

Receives message from MDT

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Receives a message from an MDT and displays it on Client.	TEXT_FROM_MD T
MDT_ID	X	IDENT	Identifies the origin of the “text” by the transmitting device.	
TEXT	X	STRING	Free format text from transmitting device.	

7.14.1.2 Scheduler returns Client initiated message

Scheduler Send message to MDT

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Sends a message to an MDT.	TEXT_TO_MDT
SHIFT_ID		IDENT	Identifies the destination of the “text” by ident of the receiving Shift.	
MDT_ID	X	IDENT	Identifies the destination of the “text” by the ident of the receiving device.	
MESSAGE_ID	X	IDENT	Identifies the message. This ident will be used by the recipient (Client) to return transmission-status updates.	
TEXT	X	STRING	Free format text to be displayed on the receiving device.	
spare		STRING	May be used to extend TEXT_TO_MDT without causing incompatibility with previous versions	
spare				
spare				

7.14.1.3 Scheduler response to Client initiated Job despatch

Scheduler Send Job-details message to MDT (Sheet 1 of 2)

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Sends a Job-details message to an MDT.	JOB_TO_MDT
SHIFT_ID	X	IDENT	Identifies the Shift the Job is to be despatched to.	
MDT_ID	X	IDENT	Identifies the destination of the “text” by the ident of the receiving device.	
MESSAGE_ID	X	IDENT	Identifies the message. This ident will be used by the recipient (GateWay) to return transmission-status updates.	
RUN_ID		IDENT	Identifies the run the Job or Depot-Stop belongs to.	
JOB_ID	X	IDENT	Identifies the Job that is being despatched.	
JOB_CODE		IDENT	Identifies the Job being sent.	
JOB_TEXT		STRING	A text-field providing the driver with information about the Job.	

Scheduler Send Job-details message to MDT (Continued) (Sheet 2 of 2)

Field Name	M	Format	Description	Range
STOP_ID_1	X	IDENT	Identifies the first Stop that is being despatched, or the Depot-Stop id.	
STOP_CODE_1		IDENT	Code for the first Stop, or the Depot-Stop id.	
STOP_TEXT_1-1		STRING	text-fields providing the driver with information specifically for the first Stop	
STOP_TEXT_1-2				
STOP_TEXT_1-3				
STOP_TEXT_1-4				
STOP_TEXT_1-5				
STOP_TEXT_1-6				
STOP_TEXT_1-7				
STOP_TEXT_1-8				
STOP_ID_2		IDENT	Identifies the second Stop that is being despatched.	
STOP_CODE_2		IDENT	Code for the second Stop.	
STOP_TEXT_2-1		STRING	text-fields providing the driver with information specifically for the second Stop	
STOP_TEXT_2-2				
STOP_TEXT_2-3				
STOP_TEXT_2-4				
STOP_TEXT_2-5				
STOP_TEXT_2-6				
STOP_TEXT_2-7				
STOP_TEXT_2-8				
EXPECTED_JOBS		CARDINAL	The expected number of jobs for the remaining in the day.	>= 0
spare		STRING	May be used to extend JOB_TO_MDT without causing incompatibility with previous versions	
spare				
spare				

Note: Note: The JOB_TO_MDT packets definition caters for Jobs of up to 2 Stops only. For Jobs with a larger number of Stops multiple individual messages must be sent.

7.14.2 Resource Status updates from MDT

Packets under this section advise the Allocator and Oracle Real-Time Scheduler about each Resource's progress.

Resource (Arrived/Started/Completed) Updates from MDT

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	The Resource has arrived at the specified Stop.	MDT_ARRIVED
			The Resource has started the specified Stop.	MDT_STARTED
			The Resource has completed the specified Stop.	MDT_COMPLETED
MDT_ID	X	IDENT	Identifies the originating driver/ Resource.	
JOB_CODE		IDENT	Identifies the Job.	
STOP_CODE	X	IDENT	Identifies the (Depot)Stop.	
DATE		DATE	The date and time of day at which the event occurred.	yyyymmdd hhmm [now] If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
TIME		HHMM		

The following two packets notify Oracle Real-Time Scheduler that the Resource has arrived at the Depot in order to load, or has actually started loading. Oracle Real-Time Scheduler does not require these packets to be sent, but they may be useful to provide accurate timing of events.

On receipt of such a packet, Oracle Real-Time Scheduler will identify the first matching Depot-Stop, adjust its time (and possibly position) and set its status to ARRIVED/

STARTED. If necessary, Oracle Real-Time Scheduler will also adjust the position of the load.

Resource (Arrived/Started at Distribution Depot) Updates from MDT

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	The Resource has arrived at the Depot to collect the specified Job for Distribution.	MDT_ARRIVED_DIST
			The Resource has started loading the specified Job.	MDT_STARTED_DIST
RSRC_ID	X	IDENT	Identifies the Resource by IDENT.	One of RSRC_ID or MDT_ID must be supplied, but not both.
MDT_ID		IDENT	Identifies the Resource by MDT_ID.	
DEPOT_ID	X	IDENT	Identifies the Distribution-Depot at which the event occurred.	
JOB_CODE	X	IDENT	Oracle Real-Time Scheduler applies the event to the first Job with matching JOB_CODE that is CLOSED or DESPATCHED to the specified Depot and Resource.	
DATE		DATE	The date and time of day at which the event occurred.	yyyymmdd
TIME		HHMM		hhmm [now] If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))

The MDT_COMPLETED_DIST notifies Oracle Real-Time Scheduler that the Resource is about to leave (has just left) the Depot. Its operation is similar to that of the above packets, but additionally it can also update weight/volume and the COMMENT field for the Job just loaded.

Resource (Completed Distribution Depot) Update from MDT (Sheet 1 of 2)

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	The Resource has completed the specified Stop.	MDT_COMPLETED_DIST

Resource (Completed Distribution Depot) Update from MDT (Sheet 2 of 2)

Field Name	M	Format	Description	Range
RSRC_ID	X	IDENT	Identifies the Resource by IDENT.	One of RSRC_ID or MDT_ID must be supplied, but not both.
MDT_ID		IDENT	Identifies the Resource by MDT_ID.	
DEPOT_ID	X	IDENT	Identifies the Distribution Depot at which the Job was picked up.	
JOB_CODE	X	IDENT	Oracle Real-Time Scheduler applies the event to the first Job with matching JOB_CODE that is CLOSED or DESPATCHED to the specified Depot and Resource.	
DATE		DATE	The date and time of day at which the event occurred.	yyyymmdd
TIME		HHMM		hhmm [now] If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
WEIGHT		DOUBLE	The weight and/or volume of the goods Collected. If specified, and goods for only one Distribution are Collected, Oracle Real-Time Scheduler will replace the weight and/or volume for this Distribution-Stop. **Note:** Weight and volume cannot be updated when more than one load is collected.	
VOLUME		DOUBLE		
TEXT		STRING	This field, if present, will be written to the COMMENT field for all Stops belonging to the Job.	

Resource Enroute, Driver Acknowledge/ClearStatus from MDT

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	The Resource will next advance towards the specified Stop.	MDT_ENROUTE
			The driver acknowledges receipt of the specified Job	MDT_ACK
			The driver wishes to revert back to the default status: ACKNOWLEDGED for Job-Stops, ALLOCATED for Depot-Stops.	MDT_CLRSTAT
MDT_ID	X	IDENT	Identifies the originating driver/ Resource.	
JOB_CODE		IDENT	Identifies the Job.	
STOP_CODE	X	IDENT	Identifies the (Depot)Stop.	

Stop Deferred from MDT

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	The drivers defers a Stop to no earlier than the specified time.	MDT_DEFERRED
MDT_ID	X	IDENT	Identifies the originating driver/ Resource.	
JOB_CODE		IDENT	Identifies the Job.	
STOP_CODE	X	IDENT	Identifies the Stop.	
DATE		DATE	The date and earliest time of day at which the Stop can be revisited.	yyyymmdd hhmm [never] If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
TIME		HHMM		

Resource Current Position from MDT

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Current position update from MDT	MDT_CURPOS
MDT_ID	X	IDENT	Identifies the originating Resource.	
LATITUDE	X	DOUBLE	Position of the Resource. Accurate positioning needs 6 significant digits after the decimal.	
LONGITUDE				

Resource Current Position from MDT

Field Name	M	Format	Description	Range
DATE		DATE	Date and time of day at which the Resource was at the specified position. Either both or none must be present.	yyyymmdd hhmm [now] If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
TIME		HHMM		

LogOn/Off, Taking Break, Driver Delayed from MDT

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	The driver logs on to the next Shift, or has completed a Break.	MDT_LOGON
			The driver has started the next Break.	MDT_BREAK
			The driver has been delayed until the specified time.	MDT_DELAYED
			The driver logs off from the current Shift.	MDT_LOGOFF
MDT_ID	X	IDENT	Identifies the originating Resource.	
DATE		DATE	Date and time of the event, except in case of DELAYED, where it represents the expected time of day at which the delay ends	yyyymmdd hhmm [now] If time zones are used then the range will be: ((hhmmZ-hhmm) to (hhmmZ+hhmm)). That is: ((local time - UTC offset) to (local time + UTC offset))
TIME		HHMM		

Resource Un-Delayed from MDT

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	The delayed Driver is available once again.	MDT_DELAY_COMPLETE

Resource Un-Delayed from MDT

Field Name	M	Format	Description	Range
MDT_ID	X	IDENT	Identifies the originating Resource.	

Scheduler Reporting MDT Transmission Status

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Reports on the progress of a transmission.	MSG_STATUS
MDT_ID	X	IDENT	Identifies the MDT being reported on.	
MESSAGE_ID	X	IDENT	Identifies the message being reported on.	
STATUS_MAJOR	X	TOKEN	High-level status which enables Oracle Real-Time Scheduler to alert the Allocator.	IN_PROGRESS RETRYING FAILED SUCCEEDED
STATUS_MINOR		STRING	Low-level status - detailed description.	
spare				
spare				
spare				

MDT acknowledges the receipt of the Go_home/Resume_Work instruction sent from Oracle Real-Time Scheduler. Please see 7.13.10 “Instructing Shift to GO_HOME/ RESUME_WORK” on page 7 - 142 for the relevant packets.

MDT_ACK_GO_HOME/MDT_ACK_RESUME_WORK packets

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Advises the operator that the Go_Home instruction has been received at the resource MDT.	MDT_ACK_GO_HOME
			Advises the operator that the Resume_work instruction has been received at the resource MDT.	MDT_ACK_RESUME_WORK
MDT_ID	X	IDENT	Identifies the resource by the ident of the receiving device.	
spare		STRING	Spare fields can be used to extend the MDT_ACK_GO_HOME/ MDT_ACK_RESUME_WORK without having to change the interface.	
spare				
spare				

The following two packets advice Allocator and Oracle Real-Time Scheduler about the shift status change as a result of the Go_home/Resume_Work action taken by the Shift. Please see 7.13.10 “Instructing Shift to GO_HOME/RESUME_WORK” on page 7 - 142 for the relevant packets.

MDT_GO_HOME/MDT_RESUME_WORK packets

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Advises the operator that the resource has started to move towards logoff.	MDT_GO_HOME
			Advises the operator that the resource has stopped moving towards logoff.	MDT_RESUME_WORK
MDT_ID	X	IDENT	Identifies the resource by the ident of the receiving device.	
spare		STRING	Spare fields can be used to extend the MDT_GO_HOME/MDT_RESUME_WORK without having to change the interface.	
spare				
spare				

ETA_FROM_MDT packets

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Updates the Stop’s external ETA into SmAuto from the MDT device.	ETA_FROM_MDT
MDT_ID	X	IDENT	Identifies the resource by the ident of the receiving device.	
JOB_CODE		IDENT		
STOP_CODE	X	IDENT	Identifies the Stop for which the ETA is being sent from MDT.	
ETA_DATE	X	DATE	Contains the ETA date	
ETA_TIME	X	HHMMSS	ETA time of day	If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)). That is: ((local time - UTC offset) to (local time + UTC offset))
spare		STRING	Spare fields can be used to extend the ETA_FROM_MDT without having to change the interface.	
spare				
spare				

7.14.3 Autodirect message for Depot Stop

Autodirecting Depot Stop

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Sends a Depot-details message to an MDT.	GOTO_DEPOT
SHIFT_ID	X	IDENT	Identifies the Shift the Depot Stop belongs to.	
MDT_ID	X	IDENT	Identifies the destination of the “text” by the ident of the receiving device.	
MESSAGE_ID	X	IDENT	Identifies the message. This ident will be used by the recipient (GateWay) to return transmission-status updates.	
DEPOT_TEXT		STRING	A text-field providing the driver with information about the Depot.	
DEPOT_ID	X	IDENT	Identifies the Depot Stop.	
STOP_CODE		IDENT	Code for the Depot Stop.	
STOP_TEXT_1_1		STRING	text-fields providing the driver with information specifically for the Depot Stop	
STOP_TEXT_1_2				
STOP_TEXT_1_3				
STOP_TEXT_1_4				
STOP_TEXT_1_5				
STOP_TEXT_1_6				
STOP_TEXT_1_7				
STOP_TEXT_1_8				
spare				
spare				
spare				

Arrived/Started/Completed Autodirected Depot from MDT

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	The Resource has arrived at the Depot Stop.	ARRIVED_DEPOT
			The Resource has started at the Depot Stop.	STARTED_DEPOT
			The Resource has completed at the Depot Stop.	COMPLETED_DEPOT
RSRC_ID	X	IDENT	Identifies the Resource by IDENT.	One of RSRC_ID or MDT_ID must be supplied, but not both.
MDT_ID		IDENT	Identifies the Resource by MDT_ID.	
DEPOT_ID	X	IDENT	Identifies the Depot Stop.	
STOP_CODE		IDENT	Code for the Depot Stop.	
DATE		DATE	The date at which the event occurs.	yyyymmdd
TIME		HHMM	The time at which the event occurs.	hhmm [now] If time zones are used then the range will be: ((hhmmssZ-hhmmss) to (hhmmssZ+hhmmss)). That is: ((local time - UTC offset) to (local time + UTC offset))

7.14.4 Break Despatch message to MDT

Scheduler’s response to Client initiated Break despatch message is transmitted to MDT using the BREAK_TO_MDT packet.

Break Despatch to MDT

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Sends a Break-details message to an MDT.	BREAK_TO_MDT
SHIFT_ID	X	IDENT	Identifies the Shift by ident.	
MDT_ID	X	IDENT	Identifies the Resource by MDT_ID.	
MESSAGE_ID	X	IDENT	Identifies the message. This ident will be used by the recipient (GateWay) to return transmission-status updates.	
DURATION	X	HHMM	Indicates the duration of the Shift.	hhmm
spare		STRING		
spare		STRING		
spare		STRING		
spare		STRING		
spare		STRING		

7.14.5 Autodirect message

Autodirect Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Set Autodirect parameters.	AUTO_DIRECT
MODE	X	TOKEN	ON - Enables Autodirect, OFF - Disable Autodirect.	ON, OFF
ON_START		TOKEN	Autodirect when the last stop starts, NO - Autodirect when the last stop completed.	YES, NO
CLOSE_RUN		TOKEN	Close run when Autodirect a distribution stop.	YES, NO
DIRECT_BREAK		TOKEN	Autodirect Break.	ON, OFF
TIME_HORIZON		HHMM	Time horizon for Autodirect.	hhmm

Autodirect Packet

Field Name	M	Format	Description	Range
spare		STRING	May be used to extend Autodirect without causing incompatibility with previous versions.	
spare		STRING		
spare		STRING		
spare		STRING		
spare		STRING		
spare		STRING		

7.15 SWITCH INTERFACE PACKETS

This section covers packets that are specific to Oracle Real-Time Scheduler Switch and host interface.

7.15.1 Packet destinations/delineation

With the exception of CutOff, requests or updates received by the Switch from a Data Interface process consist of a sequence of packets. In order to identify to the Switch what it intends the Switch to do with the packets the Data Interface process must bracket the messages in specific header and trailer packets.

Three sets of “brackets” have been defined: two for new Jobs and Slot requests (Data Interface process only) and one for generic data-updates (Database backup processes):

1. For new Jobs- **-JOB_START/JOB_END** packets.
2. For Slot-requests **-REQ_START/REQ_END** packets.
3. For Vehicle, Shift, Depot and Slot information **-DATA_START/DATA_END** packets.

In order to simplify processing by the recipient, the Switch brackets all batches of data being transferred to Database backup process in DATA_START/DATA_END packets.

The Switch expects data used for reloading from Database backup process to be enclosed in DATA_START/DATA_END packets.

Bracket Packets to Identify Intermediate Packet(s) purpose to Switch

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Followed by a STOP, STOP_TW, JOB_ASSIGN and JOB_END packets.	JOB_START
			Followed by a STOP, STOP_TW, CHS_SLOT, STOP_DEL and REQ_END packets.	REQ_START
			Followed by any combination of data-packets, and finally DATA_END. **Note:** *Note that any objects referred to in a packet must be sent in first.*	DATA_START
			Sequence terminators.	JOB_END
				REQ_END
				DATA_END

7.15.2 Error handling

7.15.2.1 Transmission errors

7.15.2.1.1 Transmission from host to the Switch

All groups of packets received from any of the host processes are acknowledged, either upon receipt by the Switch, or when processing is complete. When the host process does not receive acknowledgment, it re-sends the packet(s).

7.15.2.1.2 Transmission from the Switch to Database backup process

All information sent from Switch to Database backup process is terminated by a DATA_END packet. If this packet is not received, Database backup process requests transmission of the full schedule to ensure that it is up-to-date.

7.15.2.2 Data errors

Incorrect data may be detected by the Scheduler. Any such occurrence will be reported to the Allocator (by means of the GUI), written to a file, and returned in the form of ERROR-packets to the Client process which caused the error to occur. See “Packets communicating messages of type (WARNING and ERROR)” on page 7 - 111.

7.15.3 Primary Switch Initialisation

Upon start-up of the Primary Switch it awaits a connection from the Database Loading process and negotiates a load of all current Vehicle, Shift, Depot and Job information.

7.15.3.1 Negotiating Data Load

The data returned by Database Loading process is bracketed by DATA_START and DATA_END packets. See “Packet destinations/delineation” on page 7 - 159

7.15.4 Connection of SmAuto Processes

7.15.4.1 Initialisation of SmAuto Processes

As the state of the data contained within a connecting SmAuto can not be determined by the Switch, the Switch will reset SmAuto and then re-send ALL data packets.

7.15.4.2 Scheduler/Slot-generator determination

Switch confirms receipt of all data by sending a DATA_DONE packet before assigning an operating mode for the connecting SmAuto.

7.15.5 Removing Jobs

The Data Interface process can remove Jobs by sending a DATA_START, STOP_DEL, DATA_END packet sequence to the Switch. Associated time-windows will automatically be removed. The Switch will reply with DATA_DONE when the Job has been reliably (that is, from the secondary switch) removed.

7.15.6 Communication

Data Interface processes and Database backup processes communicate with the Oracle Real-Time Scheduler Primary Switch as described below.

7.15.6.1 Port

Each type of Client will connect on a port allocated to that type of Client only. Please refer companion Oracle Real-Time Scheduler Installation Manual for more details on setting runtime configuration.

7.15.6.2 Connecting to a Switch

Upon connection to Switch, all processes identify themselves to Switch using a CLIENT_TYPE packet (see “Client Identification Packet” on page 7 - 18). The Switch responds with a CONNECT packet (see “Connect/Disconnect Packet” on page 7 - 28).

Procedure for Connecting to Switch

Process (Database backup process, Data Interface process etc.)	⇔	Switch
		“Listens” for connections on an agreed port-number.
Establishes a connection with the Switch.		
Sends a CLIENT_TYPE packet (IDENT=process-name, VERSION=8_1, TYPE=GATEWAY)	⇒	Receives CLIENT_TYPE.
Waits for a reply.		
Receives reply.	⇐	Replies with CONNECT.
Commences operation.		

Note: The Switch “polls” for new connections at an interval of 5 seconds. Therefore the initial CONNECT response after connection may be delayed by up to an additional 1 seconds.

7.15.6.3 Checking Switch is Alive

To make sure the Switch is responding a PING packet is used.

Packet to Check if Switch is Alive

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Echo request.	PING
			Echo response.	PONG

Procedure to Check if Switch is Alive

Process (Database backup process, Data Interface process etc.)	⇔	Switch
Sends a PING packet	⇒	Receives PING.
Waits for a reply.		
Receives reply.	⇐	Replies with PONG.

7.15.6.4 Standard Procedures

The interaction between Scheduler, Switch and host-processes employs a number of standard procedures.

7.15.6.4.1 Update Procedure

The Update procedure is invoked by the host-process in order to change the schedule or schedule data. Confirmation is always received by way of a DATA_DONE or DATA_FAILED packet.

DATA_DONE is received, only after a response is received at the Switch from the Scheduler, even after the occurrence of an ERROR.

DATA_FAILED is received when a response is not received at the Switch from the Scheduler.

Update Procedure

Scheduler	⇔	Switch	⇔	Process (Database backup process, Data Interface process etc.)
Receives BATCH_ON/ OFF sequence and updates the schedule.	⇐	Receives the DATA_START/END sequence, converts DATA to BATCH and passes on to the Scheduler. BATCH_ON 1 or more data packets. BATCH_OFF	⇐	Sends a data-update sequence: DATA_START 1 or more data packets. DATA_END
May generate errors as part of the update. ERROR[PACKET=BATCH_ON]	⇒	Passes ERROR packets on to the originating client without further interpretation.	⇒	Receives ERROR packets. Takes appropriate action.
When completed, sends BATCH_DONE	⇒	Receives BATCH_DONE. Commences **UpdateConfirmation <BATCH_DONE>**		
			⇒	Receives DATA_DONE or DATA_FAILED.

7.15.6.4.2 Update confirmation packets

Switch sends an update confirmation response to the request originator using the DATA_DONE or DATA_FAILED packets. DATA_DONE is sent, only when a response is received at the Switch from the Scheduler, even after the occurrence of an ERROR.

DATA_FAILED is sent when a response was not received at the Switch from the Scheduler.

DATA_DONE Packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Switch sends DATA_DONE in the update confirmation response to the request originator, if it receives a response from the Scheduler.	DATA_DONE

DATA_FAILED packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Switch sends DATA_FAILED in the update confirmation response to the request originator, if it does not receive a response from the Scheduler.	DATA_FAILED
TEXT		STRING	May contain the reason for failure. For example: "Failed to receive a response from SmAuto."	

7.15.6.4.3 Request Update Procedure

The Request Update procedure represents an update to the schedule or schedule data in the shape of a request, i.e. the Scheduler is requested to perform an operation on behalf of the host process. The host process will receive a confirmation upon completion of the operation, otherwise a failure packet. The actual CONFIRM packet depends on the nature of the operation invoked by the host-process.

Request Update Procedure (Sheet 1 of 2)

Scheduler	⇔	Switch	⇔	Data Interface process
				Sends a request. Retains the request in case a re-send is required.

Request Update Procedure (Continued) (Sheet 2 of 2)

Scheduler	⇔	Switch	⇔	Data Interface process
		Sends the request to the scheduler.	⇐	<REQUEST>
Receives and processes the request packet.	⇐	<REQUEST>		
Generates an ERROR. ERROR[packet=<REQUEST>]	⇒	Receives ERROR. Sends <FAIL>. to the Client.	⇒	Receives FAIL and takes appropriate action.
OR				
Completes request. Confirms. COMPLETE[packet=<REQUEST>	⇒	Receives COMPLETE. Commences **UpdateConfirmation <CONFIRM>**		
				Receives <CONFIRM>. <CONFIRMA> is received only when the operation was successful, i.e. when no errors have occurred.

A Standard Request covers both the Conditional Job Assignment (see “Conditional Assignment of Job” on page 7 - 126) which is a part of the New Job Procedure and CutOff (see “Manually Specify Depot Cut-Off” on page 7 - 122).

Packets used for Conditional Assignment and Cut-Off

Request Type	Request Packet	Confirmation Packet	Rejection Packet
Conditional Job Assignment	JOB_ASSIGN_COND	JOB_CONFIRMED	JOB_REJECTED
Conditional Job Assignment	CA_JOB_ASSIGN_COND	JOB_CONFIRMED	JOB_REJECTED
CutOff	DEPOT_CUTOFF	DEPOT_CUTOFF_CONFIRMED	ERROR

7.15.6.4.4 Update Confirmation Procedure

The Update Confirmation procedure is used to report to a host-process that a change to the schedule or schedule data it has instigated has been copied into both the Primary and Secondary Switches. The actual CONFIRMATION packet depends on the nature of the operation invoked by the host-process.

Update Confirmation Procedure

Scheduler	⇔	Switch	⇔	Process (Database backup process, Data Interface process etc.)
Receives SYNC_PLAN, and sends generic data-update sequence.	⇐	Starts generic data-confirmation sequence: Forces immediate update of the Switch. SYNC_PLAN		Sends a request. Retains the request in case a re-send is required.
BATCH_ON .... BATCH_OFF	⇒	Receives BATCH_ON/OFF sequence Updates Switch. Updates secondary Switch. Sends confirmation to Client.		
		Sends <CONFIRM>	⇒	Receives <CONFIRM>.

7.15.6.5 Job removal

Jobs can be removed by any type of Client, using the standard Update procedure.

Procedure for Removing Job

Scheduler	⇔	Switch	⇔	Process (Database backup process, Data Interface process etc.)
Receives BATCH_ON/OFF sequence and removed the specified Job from the schedule.	⇐	Receives the DATA_START/END sequence, converts DATA to BATCH and passes on to the Scheduler. BATCH_ON STOP_DEL BATCH_OFF	⇐	DATA_START STOP_DEL DATA_END
May generate errors as part of the update. ERROR[PACKET=BATCH_ON]	⇒	Passes ERROR packets on to the originating client without further interpretation.	⇒	Receives ERROR packets. Takes appropriate action.
When completed, sends BATCH_DONE	⇒	Receives BATCH_DONE. Commences **UpdateConfirmation <BATCH_DONE>**		
			⇒	Receives DATA_DONE or DATA_FAILED.

7.15.6.6 Data Interface process

7.15.6.6.1 New Job Procedure

A new Job is sent as a JOB_START, Stop, Stop_TW, JOB_ASSIGN_COND, JOB_END sequence. The Switch Breaks the sequence up into a procedure similar to an Update (see “Update Procedure” on page 7 - 163), followed by a standard RequestUpdate procedure (see “Request Update Procedure” on page 7 - 164). The first procedure adds the Job information into the Scheduler, the second requests the Scheduler to assign the Job to an appropriate Shift, if possible.

Upon failure of the procedure, the Data Interface process must start an Update procedure to remove the Job from the Scheduler.

Procedure to Add a New Job from Host System (Sheet 1 of 2)

Scheduler	⇔	Switch	⇔	Data Interface process
				Sends a new Job to be assigned. Retains the Job in case a re-send is required.
		Receives the JOB_START, JOB_END sequence. Sends the actual Job to the Scheduler. Sends the JOB_ASSIGN_COND to the scheduler.	⇐	JOB_START STOP STOP_TW JOB_ASSIGN_COND JOB_END
Receives and stores the Job. Sends errors associated with the Job itself.	⇐	BATCH_ON STOP STOP_TW BATCH_OFF JOB_ASSIGN_COND		
May generate errors as part of processing the Job. ERROR[packet=batch_on]	⇒	Passes ERROR packets on to the originating client without further interpretation.	⇒	Receives ERROR packets. Takes appropriate action.
Sends confirmation. BATCH_DONE	⇒	Receives and ignores BATCH_DONE.		
RequestUpdate <JOB_CONFIRM, JOB_REJECT> triggered by the JOB_ASSIGN_COND request		Succeeds. Sends JOB_CONFIRM.	⇒	JOB_CONFIRM. Confirms to customer.
		Fails. Sends JOB_REJECT.	⇒	Receives JOB_REJECT packets. Takes appropriate action.
				In case an ERROR or JOB_REJECT has been received, the Client initiates an **Update** procedure to remove the Job from the Scheduler:

Procedure to Add a New Job from Host System (Continued) (Sheet 2 of 2)

Scheduler	⇔	Switch	⇔	Data Interface process
			⇐	DATA_START STOP_DEL DATA_END

If an error occurs in the Scheduler during processing of the Job, the procedure will not abort. The JOB_ASSIGN_COND packet ***will*** be processed, but most likely fail and also generate an ERROR, resulting in a JOB_REJECT.

7.15.6.6.2 Slot-request

Procedure for Slot request

Data Interface process	⇔	Switch
Sends a Slot-request:		
REQ_START STOP STOP_TW CHS_SLOT STOP_DEL REQ_END	⇒	Selects a Slot-generator to generate the Slots and passes the request on.
		Receives a success/failure response from the Slot-generator:
Passes the returned Slots on to the request generator.	⇐	Success: Returns a number of Slots: CHS_REPLY [0 - 112] CHS_REPLY_END
Passes an error to the request generator.	⇐	Failure: Returns a single ERROR packet.

7.15.6.7 Database backup process

Database backup process are batch processes which connect to Oracle Real-Time Scheduler Switch perform an operation and disconnect. It is principally used to:

- obtain the latest version of schedule data, including the current Job to Shift assignments.
- update a schedule when Jobs have been moved from one Depot to the other, and the two Depots are not within the same schedule.
- reload Oracle Real-Time Scheduler with operational data either at initial start-up, or after a Catastrophic Failure.
- add Resources for the 21st day and
- remove Resources and Jobs completed the day before

7.15.6.7.1 Update schedule information

Database backup process will regularly connect to the primary Switch to obtain the latest set of updates. The updates are used to maintain a recent image of the schedule in the host Data Base, which in turn may be used to reload the schedule after a Catastrophic Failure, or to generate statics. In order to minimise network traffic and effort required to update the Data Base, the Switch will usually send only those packets required to bring the Data Base up-to-date. If, however, Database backup process initiates transmission with a SEND_PLAN packet (see “Request for Scheduler to send data” on page 7 - 53), or a Switch FailOver has taken place, the Switch will send a complete schedule.
The Switch will indicate that it is sending a complete schedule by inserting a ALL_DATA packet immediately after the DATA_START packet.

Schedule Update Procedure: Database backup process <=> Switch (Sheet 1 of 2)

Database backup process	⇔	Switch
Connects to schedule ***X*** and sends a SYNC_PLAN or SEND_PLAN packet.		
	⇒	Checks which packets need updating and sends:
Updates Data Base using the received packets.	⇐	COST_BATCH_ON[1]
		COST_ITEM
		COST_BATCH_OFF[1]
		DATA_START [1]
		ALL_DATA[1] (*only when sending complete schedule*)
		SLOT
		SLOT_DEL
		DEPOT
		DEPOT_TW
		DEPOT_TW_DEL

Schedule Update Procedure: Database backup process <=> Switch (Continued) (Sheet 2 of 2)

Database backup process	⇔	Switch
Updates Data Base using the received packets.	⇐	DEPOT_DEL
		PARAM_DEPOT
		PARAM_DEPOT_TW
		RSRC
		RSRC_DEL
		PARAM_RSRC
		SHIFT
		BREAK [0 - 1]
		SHIFT_DEL
		BREAK_DEL
		STOP
		STOP_DEL
		STOP_TW
		STOP_TW_DEL
		PARAM_SHIFT
		PARAM_STOP (TYPE=LGON; logon Stop)[1]
		PARAM_STOP (TYPE=DIST; Stop at Depot)
		PARAM_STOP (TYPE=DROP; delivery)[1 -]
		PARAM_STOP(TYPE=BREAK; Break details)
		PARAM_STOP (TYPE=LGOF; logoff Stops) [1]
		PARAM_DEPOT
		SPEED_TW_PROFILE
		DATA_END [1]
All packets may appear [0 -] times, unless indicated otherwise. Indented packets appear in a sub-sequence with the previous less indented packet.		
Disconnects upon receipt of the DATA_END packet.		

7.15.6.7.2 Move Jobs between schedules

Database backup process are also involved in transferring Jobs between schedules. After following the sequence of “Update schedule information” on page 7 - 170” it will check for Jobs that are not assigned to a Shift. These are assumed to be Jobs that must be transferred to another schedule.

Procedure to Move Jobs b/w Schedules (Sheet 1 of 2)

Database backup process	⇔	Switch
Connects to schedule ***X*** and sends a SYNC_PLAN or SEND_PLAN packet.		
	⇒	Checks which packets need updating and sends:
receives DATA_START Updates Data Base using the received packets.	⇐	DATA_START [1] See “Update schedule information” on page 7 - 170
	⇐	DATA_END [1]
Selects all Jobs that have not been assigned to a Shift.		
Sends a STOP_DEL for all Jobs not assigned to a Depot in schedule X. DATA_START STOP_DEL [1-.] DATA_END	⇒	**Update Procedure** for every STOP_DEL sequence. “Update” procedures will effectively be executed in parallel.
receives DATA_END		
	⇐	For each received STOP_DEL sequence sends. ERROR DATA_DONE or DATA_FAILED.
Sends all Jobs assigned to a Depot in schedule X. DATA_START STOP[1-.] STOP_TW[1-.] DATA_END	⇒	**Update Procedure** for every STOP sequence. “Update” procedures will effectively be executed in parallel.
	⇐	For each received STOP_DEL sequence sends. ERROR DATA_DONE or DATA_FAILED.

Procedure to Move Jobs b/w Schedules (Continued) (Sheet 2 of 2)

Database backup process	⇔	Switch
Disconnects upon receipt of the ***LAST*** DATA_DONE or DATA_FAILED packet.	⇐	

7.15.6.7.3 Database Loading process:

Database Loading process reloads Oracle Real-Time Scheduler with operational data either at initial start-up, or after a Catastrophic Failure. Therefore Database Loading process does not need to send delete packets.

Reload Operational Data from Database Loading process to System (Sheet 1 of 2)

Database Loading process	⇔	Switch
Connects and sends all data.	⇒	Invokes an **Update Procedure** to process the incoming packets.
COST_BATCH_ON[1]		
COST_ITEM		
COST_BATCH_OFF[1]		
DATA_START [1]		
SLOT		
DEPOT		
DEPOT_TW		
PARAM_DEPOT		
RSRC		
SHIFT		
BREAK		
STOP		
STOP_TW		
PARAM_SHIFT		
PARAM_STOP (TYPE=LGON; logon Stop)*****		
PARAM_STOP (TYPE=DIST; Stop at Depot)		
PARAM_STOP (TYPE=DROP; delivery)	⇒	Invokes an **Update Procedure** to process the incoming packets.
PARAM_STOP(TYPE=BREAK; Break details)		
PARAM_STOP (TYPE=LGOF; logoff Stops)*****		
SPEED_TW_PROFILE		
DATA_END [1]		
All packets may appear [0 -] times, unless indicated otherwise.		
*****) Not required, may be omitted.		

Reload Operational Data from Database Loading process to System (Continued) (Sheet 2 of 2)

Database Loading process	⇔	Switch
Receives ERROR(s), takes appropriate action.	⇐	Sends an ERROR packet for every individual error detected. ERROR
Disconnects upon receipt of the DATA_DONE or DATA_FAILED packet.	⇐	Confirms completion with a DATA_DONE or DATA_FAILED packet. DATA_DONE or DATA_FAILED

7.15.6.7.4 Upload Oracle Real-Time Scheduler with 21st day data

DBPlanner’s 21st Day Data Upload: Procedure

Database backup process	⇔	Switch
Connects and sends all data for day 21.		
DATA_START [1]	⇒	Invokes an Update Procedure to process the incoming packets.
SLOT		
DEPOT		
DEPOT_TW		
RSRC		
SHIFT		
BREAK		
DATA_END [1]		
All packets may appear [0 -] times, unless indicated otherwise.		
Receives ERROR(s), takes appropriate action.	⇐	Sends an ERROR packet for every individual error detected. ERROR
Disconnects upon receipt of the DATA_DONE or DATA_FAILED packet.	⇐	Confirms completion with a DATA_DONE or DATA_FAILED packet. DATA_DONE or DATA_FAILED

7.15.6.7.5 Remove the previous day’s data from Oracle Real-Time

Scheduler

DBPlanner’s Removing Previous Day Data: Procedure

Database backup process	⇔	Switch
Connects and sends delete packets for Jobs and Resources associated with the previous day.		
DATA_START [1]	⇒	Invokes an **Update Procedure** to process the incoming packets.
SLOT_DEL		
STOP_DEL		
SHIFT _DEL		
RSRC_DEL		
DEPOT_DEL (if the Depot is no longer in use)		
DATA_END [1]		
All packets may appear [0 -] times, unless indicated otherwise.		
Receives ERROR(s), takes appropriate action.	⇐	Sends an ERROR packet for every individual error detected. ERROR
Disconnects upon receipt of the DATA_DONE or DATA_FAILED packet.	⇐	Confirms completion with a DATA_DONE or DATA_FAILED packet. DATA_DONE or DATA_FAILED

7.16 PRINT MANIFEST SERVER PACKET

Upon receiving the PRINT_RUN packet the Print Manifest Server will generate a single file with the manifests for all Runs specified in the RUN_ID field appended one after the other.

PRINT_RUN packet

Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Requests a Run be printed.	PRINT_RUN
RUN_ID	X	STRING	Comma separated list of Run Idents for which the printed Manifest is required.	
LOCATION	X	STRING	File name to write Manifest to.	
spare			May be used to extend functionality.	
spare				
spare				

Upon receiving the PRINT_SHIFT packet the Print Manifest Server will generate a single file with the manifests for all SHIFTs specified in the SHIFT_ID field appended one after the other.

PRINT_SHIFT packet

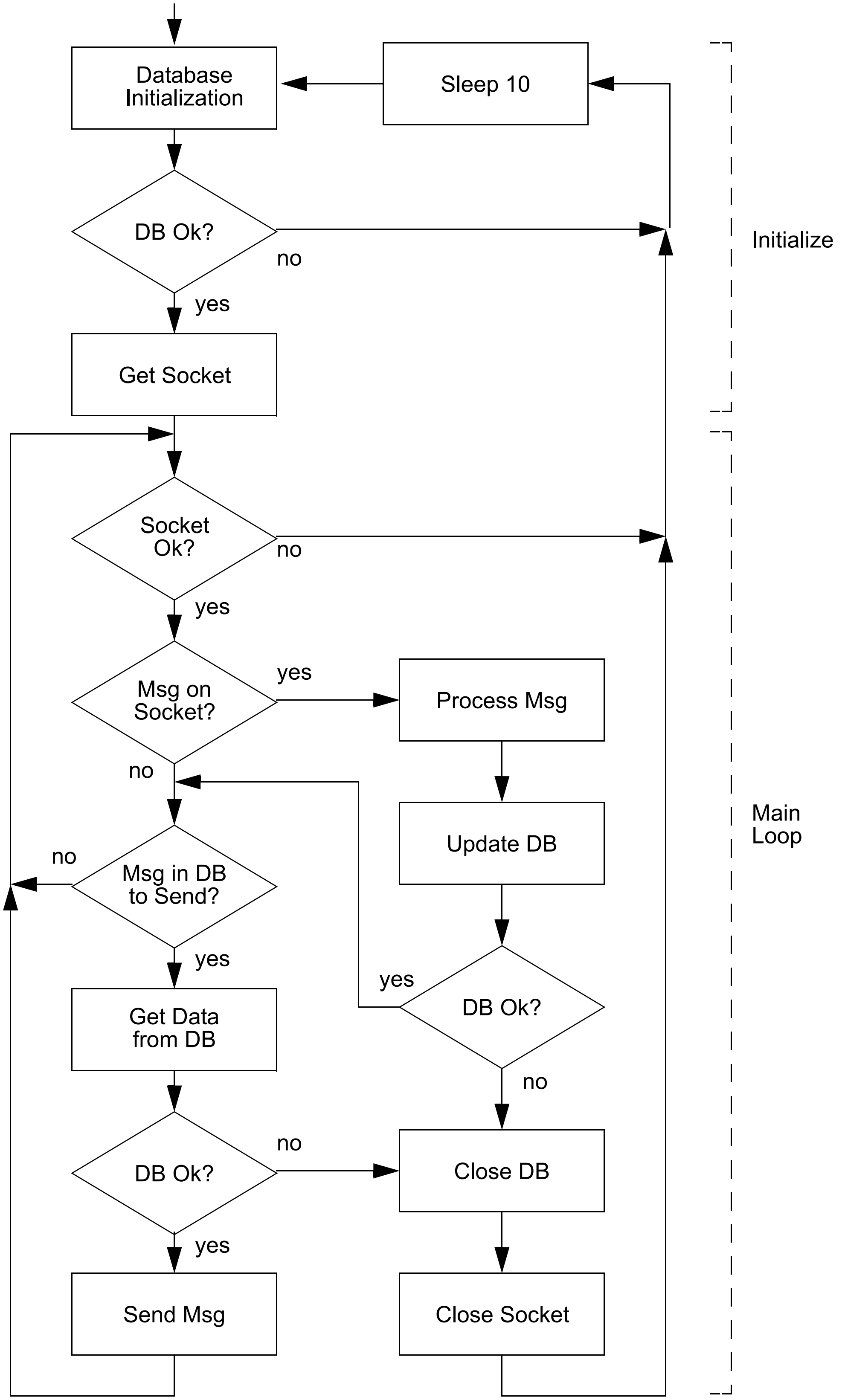
Field Name	M	Format	Description	Range
PACKET_NAME	X	TOKEN	Requests a Shift be printed.	PRINT_SHIFT
SHIFT_ID	X	STRING	Comma separated list of Shift Idents for which the printed Manifest is required.	
LOCATION	X	STRING	File name to write Manifest to.	
spare			May be used to extend functionality.	
spare				
spare				

The Print Manifest Server uses the information within the PRINT_RUN/PRINT_SHIFT packets to generate a single output file in either postscript or plain text format (The output format is specified within a configuration parameter). The output file will contain an ordered listing of the manifests for all Runs/Shifts specified in the RUN_ID/SHIFT_ID field respectively. The destination and name of the output file must be specified in the LOACTION field.

Appendix A: - Programming Examples

This section will explain the general concept of communicating with Planner in a flow-chart. The flow-chart describes the communication procedures between Gateway and Planner.

Because of the “asynchronous” nature of the communication, these examples assume that the data base has been set up to include a “transmission flag” which is used to indicate that a transmission to Planner needs to be sent, and which will be cancelled after transmission. Other “transmission status” flags may also be necessary. It is also assumed that the data base uses a “record lock” strategy to prohibit overwriting of the data.


Database Initialization
Sleep 10
DB Ok?
no
yes
Get Socket
Initialize
Socket Ok?
no
yes
Msg on Socket?
yes
Process Msg
no
Update DB
Main Loop
no
Msg in DB to Send?
yes
yes
DB Ok?
Get Data from DB
no
DB Ok?
no
Close DB
yes
Send Msg
Close Socket

A.1 COMMUNICATION WITH PLANNER

These examples typify a “C” program in a UNIX environment. To operate a Gateway in other environments, please refer to the system and programming manuals that are relevant for the particular environment.

The following program examples illustrate the code necessary to establish communication with a dedicated TCP/IP socket, to transfer Data to Planner and to send and receive messages.

A.1.1 Establishing Communication

The communication “server” software for the dedicated socket exists within Planner, so all that is necessary is to create the Gateway socket and connect it to the server.

To set up the communication channel to Planner on port number XXXX (agreed during Planner installation):

```
char                  hostname[40];
struct sockaddr_in    serv_addr;
struct hostent        * hsp;


/* get the name of the remote host */
/* which is running the socket server */


gethostname (hostname, sizeof (hostname));


/* get the network connection information about that host */
if ((hsp = gethostbyname (hostname)) == FALSE) exit;


/* set up the network communication for the specified port*/
memset ((char *) &serv_addr, 0, sizeof (serv_addr));
memcpy (&(serv_addr.sin_addr.s_addr),
hsp->h_addr, hsp->h_length);
serv_addr.sin_family = hsp->h_addrtype;
serv_addr.sin_port = XXXX;
/* now enter the connect and test loop */
while (1)
{
  fprintf (stderr, “connecting to server\n”);
  while (2)
  {
     if (initializeDB != DB_OK)
     {
        printf (“cannot initialize database”)
        exit (0);
     }
     if ((sockfd = socket (AF_INET, SOCK_STREAM, 0)) < 0)
     {
        printf (“cannot open stream socket\n”);
        exit (0);
     }

     if (setsockopt (sockfd, SOL_SOCKET, SO_REUSEADDR,
                    &on, sizeof (int)) < 0)
     {
        printerror (“error setting socket options”);
        exit (0);
     }

     if (connect (sockfd, (struct sockaddr *) &serv_addr,
                    sizeof (serv_addr)) < 0)
     {
        close (sockfd);
        fprintf (stderr, “ ...\n”);
        sleep (2);
        continue;
     }
     break;
  }
  fprintf (stderr, “connected\n”);

  if (ioctl (sockfd, FIONBIO, &nonBlockMode) != 0)
  {
     printerror (“ioctl error sockfd”);
     exit (0);
  }

     /* now test for sending and receiving messages */

  your_prog_main_loop();

     /* if you reach here, communication has been broken */

  fprintf (stderr, “lost connection with server!\n”);
  close (sockfd);
  sleep (1);
}
exit (0);
```

A.1.2 Polling for Messages

Once the socket is established, the only task is to check for incoming messages from Planner, and to send any necessary data to Planner.

This example illustrates the user code necessary to implement communication with Planner. It is an endless loop that performs both the sending and receiving tasks required when communicating with Planner.

A.1.2.1 Main Loop

Set up a loop, testing for messages from Planner and checking the data-base to see if any new information must be sent.

```
/* The host database is assumed to be initialized before */
/* this function. */
for ( ; ; )/* loop forever . . . . */
{          /* test messages */
  while (_yourCode_socket_ok () == ERROR)
  {
     sleep (10);
     _yourCode_get_socket ();
  }
  if ((nchars = yourCode_readSock()) == ERROR)
     return;
  if (nchars > 0)
  {
     if (_yourCode_getPacketFromSocket () == ERROR) return;

     if (_yourCode_decodeSocketPacket () == FALSE)
     {
        _yourCode_showString (error);
     }
     else
     {
        _yourCode_updateDatabase();
        _yourCode_flagFutureActivity()
     }
  }
  /* various data-base activities, including user written */
  /* data base tests to determine if any new jobs, etc, */
  /* or status conditions need to be sent to Planner */
...
...
  if (_yourCode_anyNewOrUpdatedDataToSend())
  {
     if (_yourCode_sendOneData() == ERROR) return;
     continue;
  }
...
}
```

INDEX