

**Oracle Real-Time Scheduler and Oracle  
Utilities Mobile Workforce Management**  
Scheduler Configuration Addendum

September 2014

Copyright © 2000, 2014 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

# Contents

---

## Addendum A

<b>Scheduler Configuration .....</b>	<b>A-1</b>
Scheduling Basics.....	A-1
The Scheduling Process .....	A-1
Request Types.....	A-2
Scheduling Parameters.....	A-2
Speed Profiles .....	A-3
Real Time Traffic .....	A-3
Auto Dispatch .....	A-3
Managing the Scheduling Process.....	A-5
Using Multiple Schedulers to Scale the Scheduling Process.....	A-5
Monitoring Schedulers .....	A-7
Failover and Load Balancing.....	A-7
Scheduler Areas.....	A-8
Multiple Service Areas within a Scheduler Area .....	A-8
Using Optimization Areas .....	A-9
Scheduler Configurations .....	A-13
Cost Profiles.....	A-13
Using the MapEditor .....	A-13
Setting Up Schedulers .....	A-13
Scheduler Registry .....	A-14
Scheduler Related Algorithms .....	A-15

## Addendum B

<b>Scheduler Cost Controls.....</b>	<b>B-1</b>
Overview of Scheduler Parameters.....	B-1
Business Drivers Related to Cost Control Settings.....	B-2
Operations Based Parameter Configurations.....	B-3
Field Service Based Operations .....	B-3
Transportation Based Operations .....	B-5
Understanding Cost Control Parameters.....	B-6
Cost Parameter Types.....	B-6
Cost Control Parameter Descriptions .....	B-10
Allocation Cost.....	B-10
Late Break Cost .....	B-10
Cumulative Capacity 1- 5 Cost.....	B-10
Complex Activity Span Cost .....	B-11
Cost for a Non-Preferred Contractor .....	B-11
Depot SLA Window .....	B-11
Depot Late .....	B-11
Depot Window Capacity 1 - 5.....	B-12
Shift Cost.....	B-12
Idle Time Cost .....	B-12
Job Attribute .....	B-12
Tasks Limit.....	B-13
Over Skill Cost.....	B-13
Life Span.....	B-13

Overtime Cost .....	B-13
Arrival Costs .....	B-14
Reserve Capacity Cost .....	B-14
Shift Area Cost .....	B-14
Shift Area Time Cost .....	B-14
No Task In Service Area Cost .....	B-15
Resource Attribute Cost .....	B-15
Depot Run Cost .....	B-15
Depot Run Separation .....	B-15
Service Class Cost .....	B-15
Shift Promotion Cost .....	B-16
Cost for a Non-Preferred Shift Class .....	B-16
Site Cost .....	B-16
Stop Promotion Factor .....	B-17
SLA Window Cost .....	B-17
Late Cost .....	B-17
Travel Distance Cost .....	B-17
Travel Time Cost .....	B-17
Under Scheduled Visit Cost .....	B-18
Wait Time Cost .....	B-18
Wait Time Past Cost .....	B-18
Window Cost .....	B-18
Service Area Cost .....	B-18
Entity Parameters .....	B-19
Cost Idle .....	B-19
Shift Distance Cost .....	B-19
Relative Late Cost .....	B-19
Relative Overtime Cost .....	B-19
Relative Efficiency .....	B-20
Relative Speed .....	B-20
Relative Shift Cost .....	B-20
SLA Flexibility .....	B-20
Preferred Time Factor .....	B-21
Relative Shift Promotion Cost .....	B-21
Relative Travel Time Cost .....	B-21
Relative Window Cost .....	B-21

## Addendum C

<b>Scheduler Parameters .....</b>	<b>C-1</b>
Parameter Descriptions .....	C-1
Appointment Generation Parameters .....	C-2
Connection Parameters .....	C-4
General Administration Parameters .....	C-5
Logging Parameters .....	C-5
Map Configuration Parameters .....	C-8
Optimizer Behavior Parameters .....	C-10
Optimizer Performance Parameters .....	C-15
Real Time Parameters .....	C-23
Reference Time Parameters .....	C-26
Scheduler Manager Parameters .....	C-27
Site Parameters .....	C-28

# Addendum A

---

## Scheduler Configuration

The scheduling process automatically assigns tasks to crew shifts, matching field work requirements to crew capabilities and optimizing the routes traveled by crews. The goal of the scheduling process is to produce the most cost effective schedule or the most efficient schedule.

This section describes how to configure the scheduling component and includes the following topics:

- [Scheduling Basics](#)
- [Managing the Scheduling Process](#)
- [Scheduler Areas](#)
- [Scheduler Configurations](#)
- [Using the MapEditor](#)
- [Setting Up Schedulers](#)
- [Scheduler Registry](#)
- [Scheduler Related Algorithms](#)

### Scheduling Basics

In general, the scheduler controls the following operational areas:

- **Shift Operations** - how shifts are allocated to tasks.
- **Break Operations** - when breaks, non-productive tasks, and periods of unavailability should occur.
- **Activities Operations** - how activities and appointments are handled and by which shifts.

The following sections provide a high level understanding on how the scheduler works and what components requires configuration.

### The Scheduling Process

The scheduling process has the following basic inputs:

- **Tasks:** Tasks can be field activities, breaks, non productive administrative tasks, and crew related periods of unavailability (POUs), all of which are incorporated into the schedule.

If using depot functionality, the scheduler inserts visits to the depot as tasks.

- **Crew Shifts:** Shift data includes crew capabilities, covered service areas and service classes, logon and logoff locations, and shift-based cost factors.

- **Contractor Capacities:** Capacity data includes covered service areas and service classes, as well as capacity amount which may be defined in number of activities or number of hours. This is only applicable if you are using capacity contractors. Refer to [Contractor Management Basics](#) for more information.
- **Depots:** If you are using depot functionality, the scheduler can manage shifts which include the location where the crew picks up and drops off goods.
- **Map Data:** All locations within the service territory are represented in the map data.
- **Static Control Data:** This includes cost controls and appointment booking groups.

Note: A user with the required authority can switch between cost profiles so that the scheduler responds and optimizes work accordingly without shutting down the scheduler.

The scheduling process assigns each task a scheduled date/time on a particular shift of a crew or a contractor capacity if using the contractors functionality. As part of the optimization process, a task may be assigned, unassigned, and reassigned any number of times until the task is dispatched. This happens when a crew logs on to the shift and the task is dispatched. Tasks may also be advance dispatched to crew shifts and contractor capacities.

The scheduler may also respond to various types of requests. Refer to [Request Types](#) for more information.

## Request Types

A scheduler that handles requests can handle the following request types:

- **Chooser** – returns a list of recommended shifts or contractor capacities for an activity
- **Appointment Booking** - returns a list of valid possible appointments for an activity.

Appointment booking refers to how the system handles requests from the host system for appointments and how it responds with available appointment windows.

The appointment booking process receives appointment requests from the host that specify the time periods for which the scheduler should return a list of available appointment windows. Refer to [Appointment Booking and Appointment Booking Groups](#) for more information.

- **Conditional Booking** - confirms the schedule of an activity when it is interfaced to the system.
- **Crew Route and Directions** – returns the street level route as well as driving directions for a list of tasks in a crew shift's schedule.

The scheduler uses a scheduling horizon to determine the number of days into the future to consider when scheduling crew shifts and activities. The scheduler uses a long term horizon to determine the number of days into the future to consider for long term requests. The long term horizon must be the same or larger than the scheduling horizon.

A scheduler configured to handle requests can also be restricted to handle short term requests only (i.e. those set within the scheduling horizon), long term requests only or both.

## Scheduling Parameters

Scheduling parameters allow you to define how the scheduler functions and how it makes scheduling “decisions”. Refer to [Addendum B: Scheduler Cost Controls](#) and [Addendum C: Parameter Descriptions](#) for a complete list of scheduler parameters.

Parameters affecting scheduling may be defined at three levels:

- Task-based parameters are defined directly on activity or at the activity type level. These are also referred to as Entity Parameters in the Parameter Definition portal.
- Shift-based parameters are defined in a shift cost profile or individual crew shift record. These are also referred to as Entity Parameters in the Parameter Definition portal.
- Global scheduler parameters are defined in a **Scheduler Configuration** record. These are set at initial implementation and will likely not change again. Refer to [Scheduler Configurations](#) for more information.

## Speed Profiles

Speed profile is used in the scheduling process to adjust calculated travel times for certain locations and during a specific time frame. For example, an implementation may be aware of holidays that close down certain roads or create traffic conditions that would affect travel times. Speed factors are defined to account for any delays in travel in that location during those days and times.

The speed profile does not impact which road or route the scheduler selects or adjust the travel speeds in real time based on the geographic area of the destination. Rather, it allows the scheduler to make a more precise travel time calculation with the expected traffic conditions taken into account.

To implement speed profile functionality:

1. Use the **Speed Profile** portal to set up daily templates to establish a default (no traffic), and additional travel factors such as rush hour, weekends, and holidays.
2. Create speed profile templates as collections of daily templates that define specific speed profiles.  
For example, the speed profile for weekday rush hour would indicate slower speeds than the profile for a week day at mid day or on the weekends. The scheduler adjusts the scheduled travel time based on the time of day indicated by the speed profile.
3. Associate the speed templates with geographic areas by creating speed profile geographic areas in the **Geographic Areas** portal and drawing the applicable map areas.  
This allows the scheduler to calculate more accurate travel times by adjusting the average travel speeds within the map area.

## Real Time Traffic

If your implementation uses a real time traffic service such as HERE Platform Traffic API, the system can be customized to make calls to the service as needed. For example you may design a custom algorithm to call the API when the crew goes enroute to a task located in certain cities for which you have real time traffic reporting coverage. Your logic may perform the call to obtain an estimated time of arrival for an enroute activity and update the assignment if reasonably different than the current estimate. This update allows the schedule to be adjusted for the remaining tasks on the shift. An element on the assignment business object captures the time update.

Refer to the documentation delivered with your real time traffic service software for information regarding how to call the service.

## Auto Dispatch

Dispatching is an action that locks a task's sequence in the shift schedule so no further optimization takes place and marks it as ready to be sent to the crew. Dispatching can be either manual (initiated by a dispatcher) or automatic (performed by the system without user initiation).

- Automatic dispatching applies only to a started shift. Until the crew starts their shift, the system does not automatically mark any task as ready to be dispatched.

- Manual dispatching applies to activities only. Breaks and POU tasks are always automatically dispatched.
- A dispatcher may manually request to dispatch an activity at any time. If a crew's shift has not started yet, then the activity remains queued for dispatch until the crew starts the shift.

You can enable or disable automatic dispatching of activities globally or for specific crew shifts or activity types.

## Setting Auto Dispatch Globally

To enable or disable auto dispatching globally:

- Set the **Enable Auto Dispatch** parameter to True or False in the scheduler configuration.  
(This is listed under Real Time parameters on the **Scheduler Configuration** portal.)
  - **True** - auto dispatching of activities is enabled globally, but can be disabled at the activity or shift level.
  - **False** - auto dispatching is disabled and cannot be enabled at the shift or activity level.

Additional scheduler parameters affecting auto dispatching include:

- Break Dispatch Mode
- POU Dispatch Mode
- Non-Productive Task Dispatch Mode
- Auto Dispatch Stability Period
- Auto Dispatch Interval
- Auto Dispatch On Completion
- Auto Dispatch Time Horizon
- Emergency Dispatch Mode

## Setting Auto Dispatch for Crew Shifts

To enable or disable auto dispatching at the crew shift level, set the following attributes for a particular crew shift or crew shift template:

- **Drip Mode:** Indicates how the system should handle dispatching of activities to the crew shift. Valid options are:
  - None:** Automatic dispatching is disabled. All scheduled activities must be manually dispatched to the crew by the dispatcher or by another system-invoked process.
  - All:** All scheduled activities are automatically dispatched to the crew and any additional activities scheduled throughout the shift are also auto-dispatched. This excludes any activities where the Auto Dispatch attribute is set to No on their Activity Type.
  - Standard:** Only a fixed number of scheduled activities are automatically dispatched to the crew at any given point of time. The fixed number is specified in the Drip Horizon. Note that if an activity's type does not allow automatic dispatching, the activity will not be assigned to shifts that are set to Standard drip mode.
- **Drip Horizon:** Number of activities that can be dispatched to a crew at any time. Valid only if Drip Mode is set to Standard.

## Setting Auto Dispatch for Activity Types

To enable or disable auto dispatching for a particular activity type:

- Set the **Auto-Dispatch** attribute at the activity type level.



- **Yes** - activities of this type will be automatically dispatched to the assigned crew.
- **No** - activities of this type will not be automatically dispatched, overriding the global or shift-based auto-dispatch settings.

**Note:** Preventing certain types of activities from being automatically dispatched is applicable only if auto-dispatch is enabled globally and the Drip Mode of the crew shift is set to All. Manual dispatching is not allowed if a shift is being drip fed tasks (Drip Mode of Standard).

## Managing the Scheduling Process

The term scheduler refers to the system component that manages a single scheduling process. An implementation typically runs at least two schedulers to provide for backup, but may run additional sets of schedulers if required for scalability purposes.

Each uniquely named scheduler is configured to manage a single scheduling process. Scheduling process responsibilities include:

- Holding - does not schedule shifts or tasks, only captures tasks and shifts that cannot be assigned to any online scheduler. Only create one scheduler of this type.
- Request Handler Only - only handles requests
- Optimization - uses the scheduler's full optimization functionality, but does not handle requests.
- Optimization and Request - handles optimization and requests.

A holding scheduler does not schedule shifts or tasks. It is designed to capture tasks and shifts that cannot be assigned to any online scheduler. One and only one holding scheduler should exist. Any task or shift attached to a holding scheduler is evidence of a scheduling error.

## Using Multiple Schedulers to Scale the Scheduling Process

The scheduling process is very complex and has high memory requirements that increase dramatically as the number of tasks being scheduled increases. Memory requirements also increase as the scheduling horizon (the number of days into the future that must be scheduled) increases. To achieve optimum performance without sacrificing scheduling efficiency, the system can be configured to use multiple schedulers at once and limit the number of tasks and shifts for which each scheduler is responsible.

Oracle provides general guidelines for calculating the optimum number of schedulers required for a particular installation:

- Define a variable, **L**, to be the maximum task load that can be scheduled by a single scheduler. 20,000 is the typical maximum.
- Determine the total number of tasks, of unique location, to be scheduled in a given scheduler area within the scheduling horizon.
- If the task count is less than L, then a single scheduler can cover the entire area. One extra scheduler will be required for backup.
- If the task count is greater than L, then the scheduler area must be covered by multiple schedulers working in parallel, and the area must be divided into smaller segments called optimization areas. Each scheduler is typically configured with a single optimization area.
- To calculate the number of optimization areas (schedulers) required for a large task count, use the following guidelines:
  - Though the typical maximum load (L) for a scheduler is about 20,000, schedulers configured with an optimization area should be designed for a smaller average load, typically 62.5% of L, which we call  $oL$ .

- The total number of schedulers required for the scheduler area can be estimated as follows, where  $n$  is equal to the expected maximum task count:

$$((n/oL \text{ rounded up to the next whole number}) * 3) - 1$$

The result is multiplied by 3 because three sets of optimization areas are typically required to manage a scheduler area with a large task count. Subtract 1 because the third optimization set can typically have fewer optimization areas than the first two sets. Please note that these rules are guidelines only; greater or fewer schedulers may be required depending on local geographies and particular scheduling requirements.

The table below shows an approximate number of schedulers required for various task counts, assuming the following base values:

- Maximum task count for a single scheduler:  $L = 20,000$
- Maximum task count for a single scheduler linked to an optimization area:  $oL = 12,500$  (62.5% of  $L$ )
- $n = \text{Task Count}$

Task Count (n) Low	Task Count (n) High	Scheduler Count	Explanation
0	20,000	2	If the load is less than the maximum task count, then one scheduler can handle the entire load, with an additional scheduler for backup.
20,000	25,000	5	$n = 25,000$ $n/oL (25,000 / 12,500) \text{ rounded up} = 2$ $2 * 3 - 1 = 5$
25,000	37,500	8	$n = 37,500$ $n/oL \text{ rounded up} = 3$ $3 * 3 - 1 = 8$
37,500	50,000	11	$n = 50,000$ $n/oL \text{ rounded up} = 4$ $4 * 3 - 1 = 11$

Thus, every physical location of an area must be covered by at least two schedulers for small task counts (typically under 20,000) or three schedulers for large task counts (typically over 20,000).

Note: The actual number of schedulers necessary may vary due to geographic constraints or particular scheduling requirements. Some geographic areas may be too large for a single scheduler to process. Determining the appropriate number of schedulers and defining optimum scheduling areas are essential to achieve maximum scheduling efficiency.

## Monitoring Schedulers

The application provides the ability to monitor schedulers using standard batch controls. This allows you to use third party enterprise management tools, if desired, to monitor schedulers as you would with any other batch processing.

Every scheduler must be configured with a unique batch control based on **M1-SM** so that monitoring can be executed. This batch control must be referenced on the scheduler record. For more information on creating these scheduler batch controls, refer to the Framework Administrator Guide under “Defining Batch Controls” and in the Framework Business Process Guide under “Batch Jobs”.

Once this is configured you can monitor schedulers using standard processing:

- Use an enterprise scheduler which shows if any scheduler submissions have failed. These can typically restart the scheduler using the tool's console.
- JMX monitoring is also available for enterprise management consoles such as Oracle Enterprise Manager or via simple JMX clients.
- View diagnostics in the **Batch Control Global View** portal.
- View logs in the **Batch Run Tree**.
- View statuses in the **Scheduler** portal in the **Scheduler List** and in the **Scheduler Runtime Details** zone.

Note that the **Scheduler** tab on **Activities** shows information about how the scheduler is handling the activity and whether or not any issues are detected. You can also access the **Service Area** portal and review the **Covering Schedulers** zone for information on the schedulers covering that service area.

### Locating the Applicable Scheduler for a Task or Shift

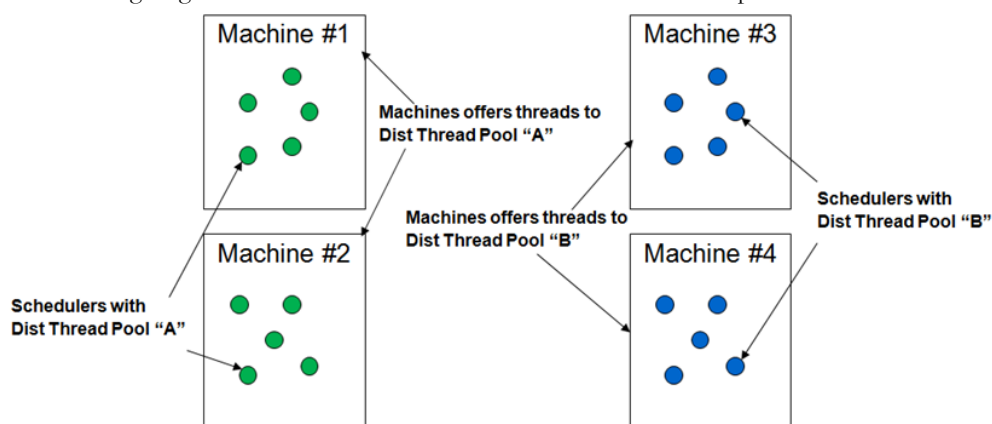
You can configure the system to provide advanced users with the ability to search for and locate schedulers that are handling a particular task or shift. The "Navigate To Scheduler Registry" BPA script can be added to your user's favorites through preferences. This script locates the related scheduler registry if any for the task or shift being displayed.

## Failover and Load Balancing

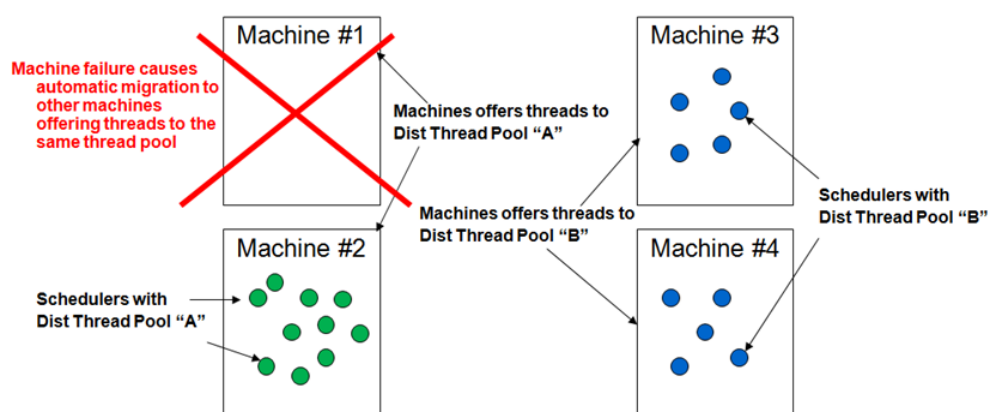
Each scheduler is controlled by a designated batch process. The batch infrastructure provides flexibility around failover and load balancing. Distributed Thread Pools allow a pool of schedulers to be associated to a pool of threads across multiple JVMs on multiple machines.

Load balancing among batch nodes assigns a thread to a JVM with the lowest percentage thread utilization for the requested thread pool. This enables active load balancing where schedulers are spread out among servers.

The following diagram illustrates scheduler distribution between thread pools and machines:



The following diagram illustrates how machine failure causes automatic migration to other machines offering threads to the same thread pool:



## Scheduler Areas

A scheduler area defines a geographic area to be scheduled by a number of cooperating schedulers. A scheduler area may comprise one or more service areas. If a scheduler area has too many tasks for a single scheduler, it may be broken down into smaller segments called optimization areas.

Refer to the online help for instructions on creating and maintaining scheduler area records. The following sections provide conceptual information to help you design your scheduler areas before entering them into the system.

### Multiple Service Areas within a Scheduler Area

A scheduler is limited to scheduling tasks within its scheduler area. Thus, the set of service areas comprising a scheduler area must at least match the most complex set of service areas covered by any given shift.

Consider a territory with eight service areas: N, E, S, W, NE, SE, NW, SW. Shifts scheduled during the day typically operate in just two service areas, while shifts scheduled during the night can be routed to any of the eight service areas:

- Day Shift (N): N, NE
- Day Shift (E): E, SE
- Day Shift (S): S, SW
- Day Shift (W): W, NW

- Night Shift: N, E, S, W, NE, SE, NW, SW

Since the scheduler area must cover the most complex set of shift service areas, the Night shift is used to model the scheduler area:

Scheduler Area: N, E, S, W, NE, SE, NW, SW.

## Using Optimization Areas

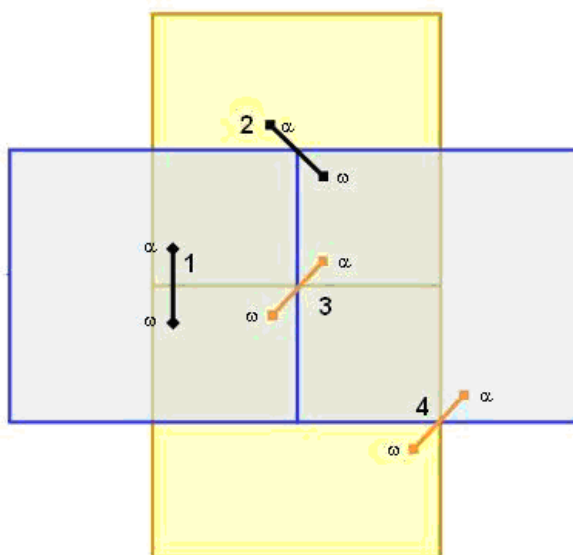
If a scheduler area has too many tasks for a single scheduler to manage (typically over 20,000), the system should be configured to run multiple schedulers in parallel and each scheduler should handle a segment of the total scheduling area. The segments, called optimization areas, should overlap each other. The overlap promotes task/shift migration between schedulers and thus achieves optimal shift routing.

### Determining the Best Optimization Area Shape

Every route that crosses an optimization area boundary represents additional workload for the scheduler. The scheduler must acquire all tasks of every shift it schedules, so it is most efficient if every scheduler has most of its shifts wholly within its own optimization area. Thus, the best shape for an optimization area is the one least likely to have routes exiting and entering the area.

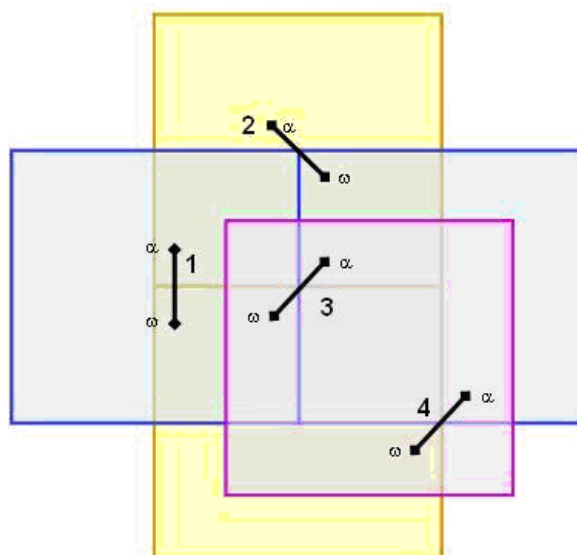
Optimization area shapes must be complementary. Optimization areas of different sets should not share common boundaries and should overlap as much as possible to promote optimum routing.

The diagram below illustrates how a pair of complementary optimization areas may be insufficient for optimal routing at specific points on the map. This simple diagram shows two optimization sets, each with two optimization areas. Each of the four optimization areas shown has its own scheduler - for simplicity, only partial sets are illustrated. The black and orange lines represent four different shifts.



Shifts 1 and 2 can be adequately covered in this configuration, because each one lies entirely within at least one scheduler. Shifts 3 and 4, however, cannot, because no single scheduler contains all the tasks of the shift. This is referred to as the corner problem, which occurs wherever the boundaries of two different optimization areas intersect and the two areas belong to different optimization sets.

Adding a third optimization set, as shown in the following diagram, solves the corner problem. All shifts can be covered as shown, because each one lies entirely within at least one scheduler.



In general, corner problems caused by optimization area overlap must be covered by a third set of optimization areas. If no corner problems exist, only two optimization sets are required.

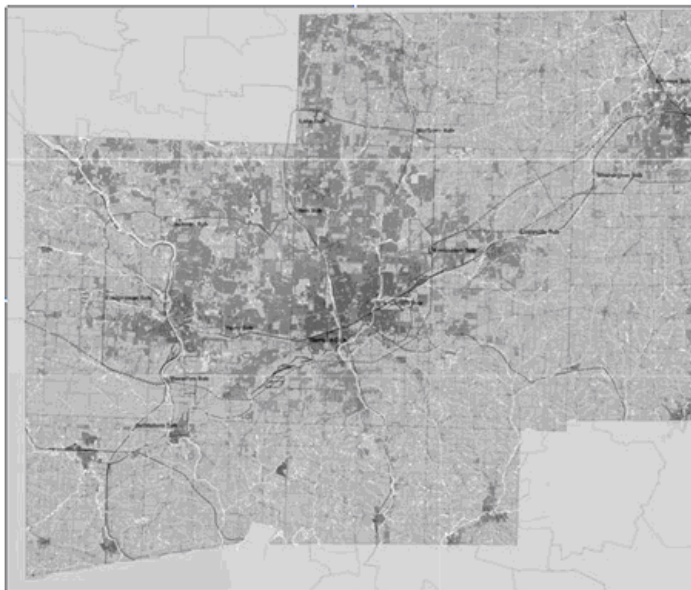
Follow these general steps when designing optimization sets:

1. Design a set of optimization areas to cover the scheduler area, where each optimization area is as large as possible (the number of tasks bounded by any area should not exceed the maximum scheduling load plus a growth factor) but still fits into a single scheduler. For best results, always try to use the least number of optimization areas possible and assign only one optimization area to a scheduler (though multiple optimization areas can be assigned to a single scheduler).
2. Design a second set of optimization areas where the optimization area boundaries are as far as possible from the first set's optimization area boundaries.
3. Design a third set of optimization areas to cover any 'corner problems' caused by the overlap of the first two sets.

#### Example:

The following image illustrates a scheduler area, named 'City', that contains more tasks than the physical limit for a single scheduler. Therefore, the scheduler area will require three sets of optimization areas.

## City



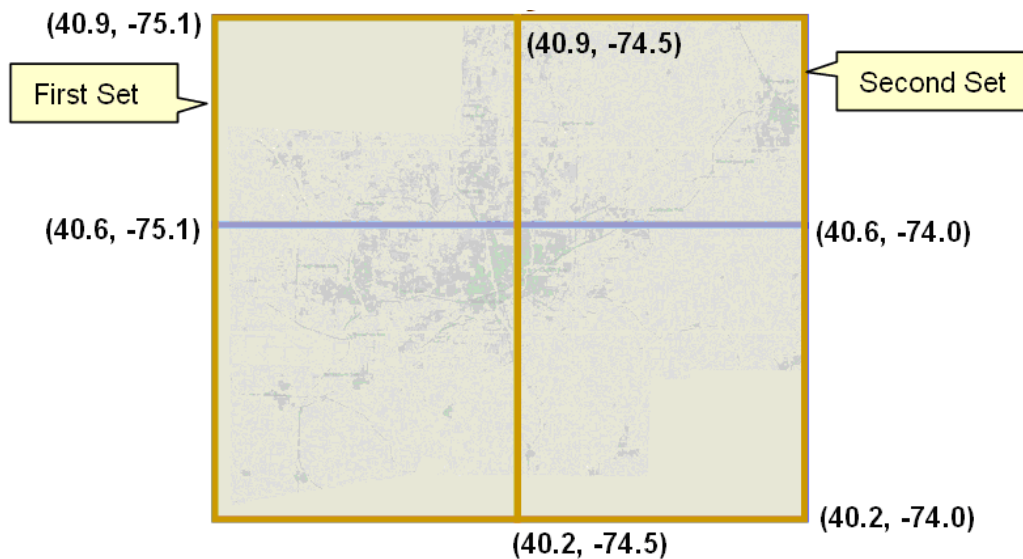
Define the first optimization set with coordinates: (40.9, -75.1 NW) to (40.2, -74.0 SE). The optimization set will have two optimization areas:

- Optimization area 1: (40.9, -75.1 NW) to (40.6, -74.0 SE).
- Optimization area 2: (40.6, -75.1 NW) to (40.2, -74.0 SE).



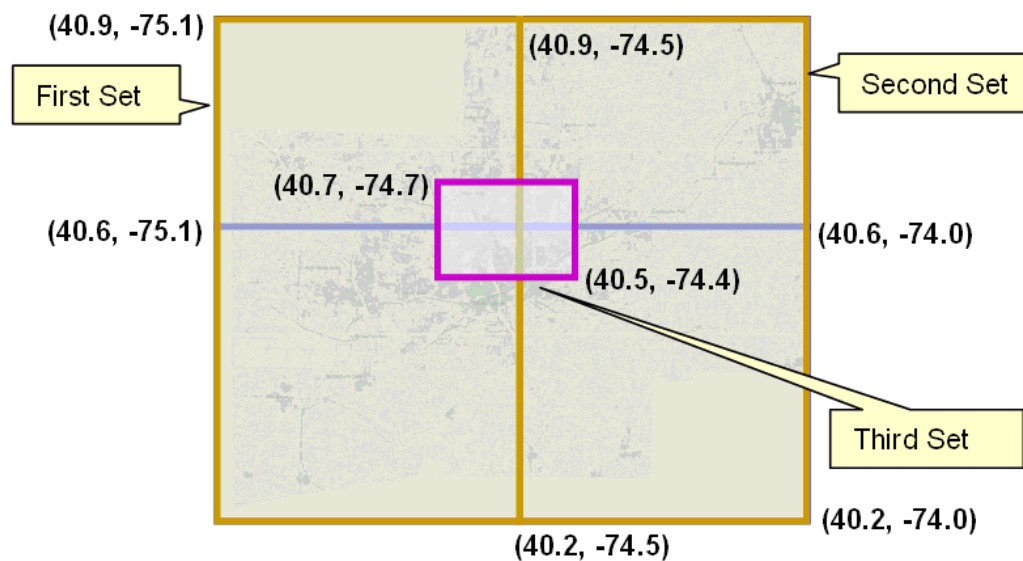
Define the second optimization set with coordinates: (40.9, -75.1 NW) to (40.2, -74.0 SE). The optimization set will have two optimization areas:

- Optimization area 1: (40.9, -75.1 NW) to (40.2, -74.5 SE).
- Optimization area 2: (40.9, -74.5 NW) to (40.2, -74.0 SE).



Define the third optimization set with coordinates: (40.7, -74.7 NW) to (40.5, -74.4 SE). The optimization set will have one optimization area:

- Optimization area 1: (40.7, -74.7 NW) to (40.5, -74.4 SE).





## Scheduler Configurations

A scheduler configuration is a predefined set of parameter values that can be associated with one or more schedulers.

Each scheduler is associated with a scheduler configuration that defines the scheduling parameters to be used by that scheduler during the scheduling process.

These parameters include cost controls, performance and optimization settings, and settings to control logging, auto-dispatching, real-time processing and other scheduler functions.

## Cost Profiles

A Scheduler Configuration record references a cost profile to define its cost parameters. A cost profile may be referenced by multiple scheduler configurations.

Cost profiles allow you to create pre-defined sets of cost controls that are configured for different circumstances. Users can actively switch between cost profiles to cause the scheduler to respond and optimize work accordingly. For example, an organization can save different profiles based on how conditions change work priorities such as: winter work, summer work, storm conditions, overtime scenarios, etc. Refer to [Understanding Cost Control Parameters](#) for more information.

A user with the required authority can switch between profiles so that the scheduler responds and optimizes work accordingly. These changes can be made without shutting down the scheduler.

Users responsible for monitoring schedulers will require training on your business practices associated to when and how to change cost profiles based.

- Use the **Scheduler Configuration** portal to add or change a scheduler configuration or a cost profile. Refer to the online help for instructions on creating a new configuration, completing it, and associating it to specific schedulers.

## Using the MapEditor

The MapEditor allows you to create map files used by the scheduler to obtain travel information and to display places and travel paths in a particular geographical region. A scheduler map consists of geographic locations, referred to as nodes, and inter-node links for which the direction, distance, and travel time are known.

When new map updates are sent from HERE maps (formerly Nokia / NAVTEQ), they can sometimes overlay the existing maps and overwrite any edits made by your implementation via the Map Editor tool. If your system is properly configured, edited maps can be reapplied onto an updated version of HERE maps.

Documentation for this tool is provided in a separate guide. Refer to the Oracle MapEditor User Guide.

## Setting Up Schedulers

This section provides a high-level summary of the steps required to set up schedulers and scheduling entities for your organization.

1. Review the scheduler-related plug-ins provided with the base system and determine if any customizations are needed. Refer to the section on Scheduler algorithms in [Appendix B: Algorithm Entities](#) for more information.
2. Use the **MapEditor** to build map files used by the scheduler to create and optimize routes, and calculate travel times and distances. Refer to the MapEditor User's Guide for details.
3. Review the scheduler parameter definitions and adjust the global default values as necessary.
4. Analyze the geographic areas requiring scheduling.  
Each scheduler is configured to cover a specific geographic area called a scheduling area. Use

the guidelines presented earlier in this section to determine number of schedulers required. If optimization areas will be used, determine optimal size and shape of each optimization area. Scheduler areas and optimization areas are defined on the Scheduler Area portal.

5. Define various cost profiles to support various circumstances as needed.
6. Determine how many scheduler configurations you must define.

**Note:** All schedulers whose scheduler area is associated with the same service area must use the same scheduler configuration. Only schedulers that do not cooperate (that is, they have different service areas) may have different scheduler configurations.

7. For each scheduler configuration, determine the scheduler parameter values to use and the number of days in the future for which data should be considered for scheduling.
8. Identify each scheduler you will be using.
9. Analyze scheduler distribution over servers as well as the necessary thread pools needed to support load balancing and failover.
10. Create a unique Batch Control record based on M1-SM for each scheduler. Ensure to associate the batch control with the appropriate thread pool name as a batch parameter.
11. Create a scheduler record and associate it with its batch control, a scheduler configuration a scheduling area and an optimization area as needed.

## Scheduler Registry

The scheduler registry contains a record for each task, shift, appointment booking group, and scheduler configuration that is linked to a scheduler. All objects associated with the scheduling process register themselves to the scheduler registry.

- An activity is registered with all schedulers that host its service area or, if optimization areas are enabled, its optimization areas. If a scheduler's scheduler area is defined using optimization areas, then the activity registers itself to that scheduler only if the scheduler covers the activity location's geocode. If a scheduler is defined using service areas only, then the activity registers itself to the scheduler if it covers its service area.
- A break is registered with all schedulers that host its shift.
- A POU task is registered with all schedulers that have the common services areas with the POU.
- A shift is registered with all schedulers that host any of its service areas.
- Every appointment booking group is registered with all schedulers.
- A scheduler configuration is registered with all schedulers that use that configuration.

The online scheduler monitoring processes update the scheduler registry automatically to remove objects that no longer belong to any of the scheduler's optimization areas (if used) or service areas (if no optimization areas are used) and to add objects that belong to one of the scheduler's optimization areas but are not already linked to the scheduler. This can happen if a scheduler's optimization areas are altered while the scheduler is running.

### Posting Actions

The system uses a posting action to communicate information to the scheduler about a change that was made to an entity. The scheduler then communicates this information to its scheduler process.

**Note:** It is the responsibility of the audit plug-in of the entity's business object to update the registry when changes are made to that entity.

Posting actions in the registry include:

- **Send:** An entity has changed and the entire entity needs to be sent to the scheduler.
- **Send Status:** An entity's state has changed and state-related details for the entity, rather than the entire entity, need to be sent to the scheduler.
- **Send Schedule:** Another scheduler manager has made a schedule change to an entity and schedule-related details need to be sent to the scheduler.
- **Processed:** The registry record has been processed by the scheduler. For example, an object that needed to be sent to the scheduler process was delivered.
- **Remove:** The system has deleted or cancelled an entity (or changed it so that it is no longer relevant to the scheduler) and the entity needs to be removed from the scheduling process (that is, it should no longer be considered for scheduling). After removing the object, the scheduler deletes the registry entry.

If any errors or warnings resulted from the posting action, they appear in the registry record. Depending on your configuration, To Do entries may also be generated for scheduler registry errors.

## Scheduler Related Algorithms

Scheduler read algorithms are used to read registered objects and package them for transfer between the scheduler manager and the scheduling process. Read algorithms are enabled for tasks, shifts, appointment booking groups, and scheduler configurations.

The base package provides several algorithms for the following scheduler-related system events. Refer to [Appendix B: Algorithm Entities](#) for more information.

# Addendum B

---

## Scheduler Cost Controls

Oracle Utilities Mobile Workforce Management and Oracle Real-Time Scheduler are heavily parameterized with the ability to support a myriad of operational use cases and scenarios. Every individual implementation will utilize a unique combination of these parameters to deliver the final operational solution.

This chapter discusses cost control and entity parameters which can be modified to best suit your business practices when configuring the system. Descriptions of the administrative scheduler parameters are provided in [Addendum C: Parameter Descriptions](#). In general these parameters should only be modified with the guidance of your Oracle representative.

For a more complete description of the scheduler in general, refer to [Addendum A: Scheduler Configuration](#).

This chapter includes the following:

- [Overview of Scheduler Parameters](#)
- [Business Drivers Related to Cost Control Settings](#)
- [Operations Based Parameter Configurations](#)
- [Understanding Cost Control Parameters](#)
- [Cost Control Parameter Descriptions](#)
- [Entity Parameters](#)

## Overview of Scheduler Parameters

Scheduler parameters control how a particular scheduler will schedule tasks, optimize routes, and dispatch activities to crews.

Parameters are typically defined as part of the implementation process. The **Parameter Definition** portal allows admin users to set the minimum, maximum, and default values for each scheduler parameter.

Scheduler parameters can be broadly categorized in two ways:

- **System and Administrative** parameters are required to control how the base application operates. As the name suggests, these are applied system wide and control general operating behavior as well as the interaction with external systems through integration. Each individual parameter setting in this group needs to be considered and mapped to the operation to ensure the correct operational behavior. System and administrative parameters are associated with a scheduler configuration and their values are maintained on the **Scheduler Configuration** portal. As mentioned, modifications to these parameters can have a very serious impact on the system and require a shut down to commit the changes.

- **Cost Control** and **Entity** parameters are required to provide detailed scheduling policies. These will differ for each implementation depending upon the type of key operations managed by an organization.

**Entity** parameters are associated with other objects in the system, such as shifts or activity types, and are maintained on the maintenance portals for those objects.

**Cost control** parameters primarily control the day today flexibility and “tuning” of the scheduler solution. These controls can be applied to the production environment without the need to bring down the system.

These controls are applied to objects that represent the operational structure, such as:

- **Activities** - data that represents tasks, order or jobs to be scheduled, but which originate in an external HOST system and are passed to the system via integration.
- **Crew Shifts** - data that represents each working day for each resource that is being scheduled by the system.
- **Depots** - data that represents locations which can be visited by the scheduled resources, and the operational controls, operational time windows and capacities for those depot locations.

## Business Drivers Related to Cost Control Settings

Typically, two major business drivers will guide your configuration of cost control parameters. These business drivers are generally tied to the areas of operation indicated. Of course, any implementation may mix and match these business goals:

- More **Efficient** Service  
Transportation Based Operations
  - Better Leverage of Workforce Skills
  - Reduction of Time Lags / Travel Waste
- More **Effective** Service  
Field Service Based Operations
  - More Responsive to Customer Requests
  - More Assurance of Timely / Complete Response

It is best not to change the Travel Distance and Travel Time costs from their default settings. These act as a reference for all other cost-controls.

# Operations Based Parameter Configurations

Most best practices for scheduler configurations fall into one of two main operations areas:

- [Field Service Based Operations](#)
- [Transportation Based Operations](#)

## Field Service Based Operations

Businesses with field service based operations typically require that a person in the field make a physical visit to the customer site. Electric, gas and water utility companies usually fall into this categorization.

Organizations with operations based mainly in the field might focus on the following considerations:

- [SLA Adherence](#)
- [Days to Serve](#)
- [Balance Planned and Reactive Work Orders](#)
- [Lowest Resource Schedule](#)

### SLA Adherence

Operations that have service level contracts to customers, generally known as service level agreements (SLA), must adhere to certain standards when making service calls. Failure to meet these standards can result in costly penalties and legal consequences.

Cost controls to configure include any lateness and time cost rules from the Shift and Activity groupings.

#### Sample of Cost Controls to Configure

- [Arrival Costs](#)
- [Late Cost](#)
- [Relative Late Cost](#) (Entity Parameter)
- [Resource Attribute Cost](#)
- SLA
  - [SLA Flexibility](#) (Entity Parameter)
  - [Preferred Time Factor](#) (Entity Parameter)

### Days to Serve

Organizations that do not have formal customer service level contracts but that maintain internal targets to offer service within a certain number of days of a customer placing a work order, offer incentives to meet these service goals.

Cost controls to configure include any time dependent cost rules, resource, shift, travel, time and shift promotion rules.

#### Sample of Cost Controls to Configure

- [Late Cost](#)
- [Shift Promotion Cost](#)
- [Preferred Time Factor](#)
- [Complex Activity Span Cost](#)

## Balance Planned and Reactive Work Orders

Some scheduler rules can help to balance situations where pre-existing planned work (such as age changes) must be superseded by higher priority reactive work order types (emergency or reconnects or credit restores). The planned work must still be completed to meet service level commitments, while the organization still completes the reactive work tasks.

When planned work orders come into the system, they come in bulk. Reactive work only comes when it occurs, so the scheduler needs to be automated to factor in reactive work. It moves planned work out to make space for reactive work while also factoring in the cost for these changes.

Cost controls to configure include any priority allocation settings, cost rules and time window automatic extension settings.

### Sample of Cost Controls to Configure

- [Shift Promotion Cost](#)
- [Stop Promotion Factor](#)
- [Preferred Time Factor](#)
- [Late Cost](#)

## Lowest Resource Schedule

In organizations where the efficiency of the operation is measured primarily on the utilization of the fewest of the available resources, shift cost and resource rules become very important. When the scheduler is set to consider the lowest resources it attempts to use fewer crews and better resources (such as varying skill levels).

Cost controls to configure include any resource (including contractors), overtime or shift cost rules.

### Sample of Cost Controls to Configure

- [Resource Attribute Cost](#)
- [Overtime Cost](#)
- [Shift Cost](#)
- [Cost for a Non-Preferred Contractor](#) - This mainly applies for **shift based contractors**. Organizations employing a number of sub-contractors with different agreements, may want the scheduler to prioritize one sub-contractor over another, depending on the Activity. A ranked list of preferred contractors is specified on the activity, any Contractor not specified is deemed to be ranked last.
- [Cost for a Non-Preferred Shift Class](#) - This only applies for organizations employing **capacity based contractors**. The same considerations apply as for shift based contractors (as described above), however organizations employing both types of contractors must also configure a "Non Preferred Shift Class" cost.
- Complex Activities require additional configuration to control the elapsed time between the first and last visit, and the duration of visits themselves.
  - [Complex Activity Span Cost](#)
  - [Under Scheduled Visit Cost](#)
  - [Maximum Overlap Between Visits](#)

## Transportation Based Operations

Business focused on transportation based operations tend to place a high value on making appointment times and maximizing traveling distances.

Organizations with operations mainly based around transportation might focus on the following considerations:

- [Appointment Compliance](#)
- [Maximum Density of Appointments](#)
- [Zero Over Capacity Schedule](#)
- [Lowest Mileage Schedule](#)
- [Lowest Resource Schedule](#)

### Appointment Compliance

In organizations where appointments are offered to ensure that deliveries and collections are made according to the appointment commitments, lateness time and distance become important factors in the scheduler configuration.

Cost controls to configure include any lateness, time or distance cost rules.

#### Sample of Cost Controls to Configure

- [Late Cost](#)

### Maximum Density of Appointments

Most organizations will need to configure the scheduler to ensure that appointments are not scheduled too close together and that there is enough time to complete work in between appointments.

Cost controls to configure include any time or distance cost rules.

#### Sample of Cost Controls to Configure

- [Late Cost](#)
- [Apply Minimum Travel Time to All Tasks](#) (Optimizer Behavior Parameter)
- [Task Grouping Travel Time](#) (Optimizer Performance)
- [Shift Promotion Cost](#) (to promote appointments/fill gaps in earlier shifts)

### Zero Over Capacity Schedule

In situations where the work load puts resources over capacity, parameters must be in place to enforce alternate scheduling policies and contingent plans to reallocate priorities.

Cost controls to configure include any weight, volume, capacity, lateness or overtime cost rules.

#### Sample of Cost Controls to Configure

- [Cumulative Capacity 1- 5 Cost](#)
- [Depot SLA Window](#)
- [Depot Late](#)
- [Depot Window Capacity 1 - 5](#)
- [Late Cost](#)
- [Overtime Cost](#)



## Lowest Mileage Schedule

Some organizations place the most value on achieving the highest efficiency in their operations by measuring the economy of scheduled routes. The scheduler should be configured to plot the shortest route possible and within the shortest amount of time.

Cost controls to configure include any time or distance cost rules.

### Sample of Cost Controls to Configure

- [Late Cost](#)
- [Task Grouping Travel Time](#) (Optimizer Performance)

## Lowest Resource Schedule

Some organizations place the most value on achieving the highest efficiency in their operations by measuring the utilization of resources. The scheduler should be configured to use the fewest and last expensive number of available resources.

Cost controls to configure include any resource, shift or overtime cost rules.

### Sample of Cost Controls to Configure

- [Resource Attribute Cost](#)
- Shift Costs - Any parameters related to maximizing shift efficiency.
  - [Overtime Cost](#)
  - [Service Class Cost](#)
  - [Shift Promotion Cost](#)

# Understanding Cost Control Parameters

The system uses cost control parameters to encourage or discourage particular actions within the system. Applying cost controls allows the system to produce the most cost-effective schedule while adhering to your business rules and processes.

Cost parameters are applied by defining a cost profile as a type of scheduler configuration then applying that cost profile to the scheduler configuration. During the course of business, schedulers can be re-configured "on the fly" in response to particular conditions (such as changes in weather or overtime requirements) by changing the cost profile on the scheduler configuration. This can be done without stopping the scheduler and the scheduler will immediately respond by optimizing work according to the changed cost profile.

## Cost Parameter Types

Cost Parameters are of two types:

- **Relative:** Cost factors that work as multipliers against global costs. For example, consider that an organization has full-time crews and contractors that may be called on to handle excess work. In order to encourage the system to schedule work to contractors only when no full-time crew is available, you could apply a relative cost of ten to contractor shifts, indicating that these shifts would cost ten times as much to use in the schedule as a full-time crew shift. Examples of relative cost parameters are Relative Shift Cost, and Relative Late Cost. Relative costs factors default to a value of 1.0.
- **Global:** Costs that are applied globally at a base level within the schedule. Examples of global cost settings are Late Cost, Shift Cost, Travel Time Cost, and Travel Distance Cost.

Global costs can be of various types:

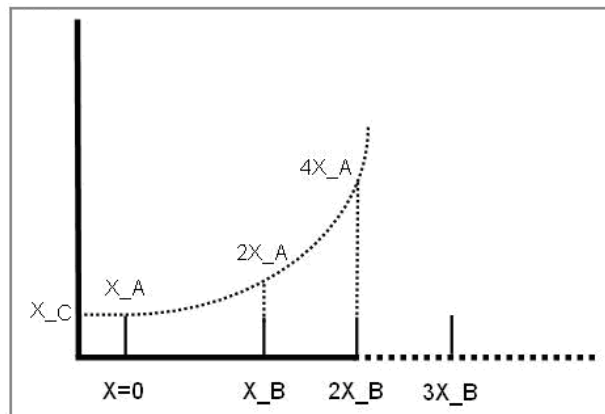
- **Flat:** Costs that are applied once, when the object is activated, and removed when the object is no longer being used. These are also known as "Standing Costs". The cost for

using this object does not go up over time but stays at a standard or flat cost. For example, the global Shift Cost is the flat cost of utilizing a shift. The Relative Shift Cost parameter can be used as a multiplier against the flat cost. Other examples of flat costs include: Shift cost, cost of a run, cost of goods, hauling costs, site costs, etc.

- **Variable:** Costs that increase linearly based on a measurable variable. These are also known as “Running Costs” or “Variable Costs”.  
For example, travel time costs may be expressed as dollars per minute or dollars per hour, and travel distance costs expressed as dollars per mile or dollars per kilometer. Other examples of rate costs include: idle time costs, access window costs, pick up or drop off services, allocation costs, site separation costs, product depot costs, etc.
- **Complex:** Costs that increase exponentially and are nonlinear in nature. The Late Cost parameter, which defines the global cost of late arrival to an activity, is a function of the number of seconds late for arrival. For example, late cost is incurred beginning with the first second the crew is late for arrival. However, the exponential function can be defined so that the late cost is insignificant for the first ten minutes, but starts to ramp up quickly thereafter. This cost pattern models the idea that the customer will accept a crew being 5-10 minutes late, but will be increasingly displeased with any further delays.

Examples of complex costs include: resource arriving late at a task, overtime, volume, weight, reserve capacity, etc.

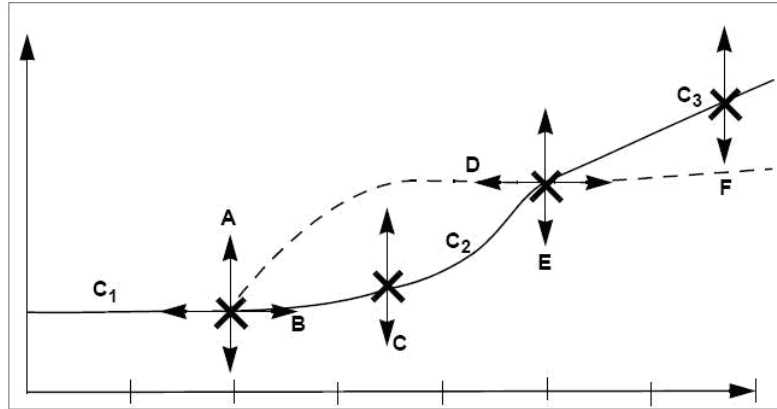
The following figure illustrates the type of curve represented by a complex cost:



Complex costs are determined by mathematical functions controlled by a group of three cost control parameters:

- Cost Value A represents the initial slope for  $x \geq 0$  (at  $x=0$ ).
- Cost Value B represents the doubling rate (the slope of the function doubles every  $x_B$ ).
- Cost Value C represents a flat additional cost, creating an initial step.
- **Time-Dependent:** Factors that adjust shift-based costs based on when a shift begins. For example, when the system is scheduling activities several days into the future, the goal is to combine activities to produce the best route for each day. As a result, some shifts may appear under-utilized or even empty. However, when scheduling activities for the current day, the main objective is to add as many activities as possible to fully utilize the crew's time. Time-dependent cost (TDC) factors allow the scheduler to produce the most cost-effective schedule considering these time factors.

To configure time-dependent costs, you must define three distinct curves.



In the diagram above, the horizontal axis is time, in day increments (Day 0, Day 1, Day 2, etc.). The initial curve is always flat. The second curve is either convex or concave. The final curve is always a straight line into infinity.

Six independent variables (TDC Values A-F) are required to define this shape:

- **TDC Value A:** Initial value for the time-dependent cost factor, which is flat from day 0 to TDC Value B. This value must be greater than or equal to zero.
- **TDC Value B:** Controls how long the initial TDC Value A applies, in hours. For example, if the initial factor should apply for all of the first day, then this should be set to 24 (hours).
- **TDC Value C:** Controls the shape of the curve between the initial value and the final slope, referred to as the transition curve. The valid range is 0.1 to 10. A value of 4 produces a curve similar to the solid line in the previous diagram.

A value of 0.2 produces a curve similar to the dashed line in the previous diagram.

A value of 1 produces a straight, linear gradient between the initial and final slope, and is a good starting point.

- **TDC Value D:** Number of hours after which the function changes from curved to linear. This value controls the width of the transition curve. This value must be greater than TDC Value B.
- **TDC Value E:** Factor at the point at which the function changes from curved to linear. (This is the final factor for the transition curve and the initial factor for the final curve.) This value must be greater than or equal to zero.
- **TDC Value F:** Controls the gradient of the final slope (units/hour).

Thus, the three curves are defined as follows:

- **Initial curve:**  $C1 (x:0..B) = A$
- **Second curve:**  $C2 (x:B..D) = ((x - B) / (D - B))C * (E - A) + A$
- **Final curve:**  $C3 (x:D..) = (x - D) * F + E$

## Time-Dependent Cost Examples

### Example 1: TDC Applied to Late Cost

Time-dependent costs can be applied to the **Late Cost** parameter to address a situation where cost should remain the same for the first two days, then rise rapidly 1 cost unit per day over the next two days and then increase by 0.2 units for all the remaining days. In this example, TDC variables would be set as follows:

A = 1

B = 48 (hours)

C = 1

D = 48 (hours)

E = 3

F = 0.0083 (0.2 / 24)

Assuming that 5 minutes of late time is costed at 100 units to begin with, applying these time-dependent costs would result in the following:

For a shift on day 1, the cost is 100

For a shift on day 2, the cost is 100

For a shift on day 3, the cost is 200

For a shift on day 4, the cost is 300

For a shift on day 5, the cost is 320

For a shift on day 6, the cost is 340

For a shift on day 7, the cost is 360

### Example 2: TDC Applied to Shift Promotion Cost

Time-dependent costs can be applied to the **Shift Promotion Cost** parameter to promote early completion of activities. The scheduler has no innate preference for activities to be completed as early as possible. Until other constraints are breached, the scheduler is mainly distance driven. Thus, it may schedule an activity for day 5, even though there is capacity on day 1, if it results in less travel. Time-dependent costing can be applied to the Shift Promotion Cost to bias against distance in favor of ensuring that no transport capacity is wasted for day 1. Ideally, costing is set so that day 1 fills up just before it becomes day 0. If costing is too weak, capacity is wasted; if too strong, days fill up well in advance, and the quality of routes suffers.

### Example 1: TDC Applied to Reserve Capacity Cost

Time-dependent costs can also be applied to the **Reserve Capacity Cost** parameter to limit the area covered for individual routes when providing appointments. This is done by invoking an additional cost when a vehicle exceeds a certain percentage (such as 50%) of its available time. Thus, the first appointment determines the general direction of that day's run, with the distance travelled for subsequent appointments being restricted through the reserve capacity cost. The more distance is being covered, the stronger the restriction, causing activities that introduce excessive travel to be deferred to another day. Time-dependent costing is required to gradually relax the Reserve Capacity Cost, enabling the system to fill day 0, day 1 and so on, in case the reserve capacity cost turns out to be too restrictive.

# Cost Control Parameter Descriptions

This section provides descriptions of the cost control parameters for reference.

To view these in the system, navigate to **Admin Menu > Scheduler > Parameter Definition** and select the filter for the **Cost Control** parameter type.

## Allocation Cost

Global cost applied to each optional task that is not assigned (i.e., on the free list).

Parameter Name	Default
ALLOCATION	5,000.0000

This parameter encourages the scheduler to fill all unused Crew capacity within the scheduling horizon with optional activities.

### Best Practices

- Start with 5000
- Unused Crew capacity that could be filled by optional Activities is not a real problem if this occurs towards the end of the scheduling horizon. If it occurs earlier, and is perceived as affecting the operation, increase the value. Do not set to excessive levels, i.e. beyond 5 x the costs set for skills or resource attributes.

## Late Break Cost

Exponential cost for taking a break late.

Parameter Name	Default
BREAK_LATE	<ul style="list-style-type: none"> <li>• Cost Value a: 1.0000</li> <li>• Cost Value b: 3600.0000</li> <li>• Cost Value c: 0.0000</li> </ul>

Note that it is not a good idea to use break late costs because they can sometimes inadvertently cause the scheduler to give breaks a higher priority than other considerations. This should be used very carefully and with guidance from an Oracle representative.

## Cumulative Capacity 1- 5 Cost

The cost of cumulative on-board capacity of the shift assigned to the activity.

Parameter Name	Default
CCAP1 - 5	<ul style="list-style-type: none"> <li>• Cost Value a: 0.0000</li> <li>• Cost Value b: 0.0200</li> <li>• Cost Value c: 0.0000</li> </ul>

## Complex Activity Span Cost

Defines costs as a function of the number of days that a complex activity visit exceeds the number of days allocated for completion.

Parameter Name	Default
COMPLEX_ACT_SPAN	<ul style="list-style-type: none"> <li>a. 5,000.0000</li> <li>b. 2.0000</li> <li>c. 0</li> </ul>

### Best Practices

- Start with 5000
- If CA's generally exceed their allotted span-time by too much, increase a.

## Cost for a Non-Preferred Contractor

This cost biases the scheduler towards scheduling an activity on a shift or capacity belonging to a preferred contractor, if specified. This mainly applies for shift based contractors.

Parameter Name	Default
CONTRACTOR_PREFHAVE	3,000.0000

## Depot SLA Window

Sets a value to indicate the cost of working a job outside the timeframe of the Depot SLA Window.

Parameter Name	Default
DEPOT_SLA_WINDOW	5,000.0000

## Depot Late

Cost as a function of the number of seconds late at a stop. In order to maintain balance with STOP\_LATE, the resulting cost is multiplied by the number of jobs transferred here.

Parameter Name	Default
DEPOTLATE	<ul style="list-style-type: none"> <li>Cost Value a: 2.0000</li> <li>Cost Value b: 1800.0000</li> <li>Cost Value c: 0.0000</li> </ul>

## Depot Window Capacity 1 - 5

Sets a value to indicate the cost of exceeding the capacity limit for a depot time window.

Parameter Name	Default
DEPWINCCAP_1 - 5	<ul style="list-style-type: none"> <li>Cost Value a: 10,000.0000</li> <li>Cost Value b: 0.0200</li> <li>Cost Value c: 0.0000</li> </ul>

## Shift Cost

Global cost of utilizing a shift. The [Relative Shift Cost](#) parameter, defined on the shift cost profile, can be used as a multiplier against this cost.

Parameter Name	Default
HAUL	0.0000

### Best Practices

- Start with 0
- If there is a cost associated with sending a crew out, shift cost may be used to minimize the number of shifts the scheduler assigns work to. Generally this means a small increase in distance travelled/activity and overtime for the shifts still being used. If used, shift cost should be in a similar range as attribute and skill costs.

## Idle Time Cost

Global cost of a crew being idle.

Parameter Name	Default
IDLETIME	0.0000

## Job Attribute

Sets a value for the cost when a Job Preference and Job Attribute do not match. For example, when transporting prisoners, it is preferable not to have hardened criminals and juveniles in the same vehicle.

Parameter Name	Default
JOB_ATTR	5000.0000

## Tasks Limit

Exponential cost for exceeding this number of tasks assigned to a shift. The input to this cost function is the number of tasks exceeding the tasks limit. For example:

- If a shift has a maximum tasks limit of 10 and the sum of all scheduled tasks is 12, then the cost input is 2.
- If a shift has a maximum tasks limit of 100 and the sum of all scheduled tasks is 102, then the cost input is also 2, so the resulting cost is the same.

Parameter Name	Default
JOBS_LIMIT	<ul style="list-style-type: none"> <li>• Cost Value a: 5000.0000</li> <li>• Cost Value b: 10.0000</li> <li>• Cost Value c: 0.0000</li> </ul>

## Over Skill Cost

Flat cost applied for each crew skill level in excess of the required skill level for an assigned task. This is used to discourage the scheduler from assigning crews that are overskilled (either in quality or in quantity) for tasks.

Parameter Name	Default
OVER_SKILL	1,000.0000

## Life Span

Sets a value to indicate the cost for late delivery. Cost is represented as a function of the number of seconds a product is on board longer than the maximum- Runlength defined for a Shift. For distribution runs, if a Shift specifies maxrunlength, and an Activity Lifespan, the shortest duration is applied.

Parameter Name	Default
LIFESPAN	<ul style="list-style-type: none"> <li>• Cost Value a: 1.0000</li> <li>• Cost Value b: 3600.0000</li> <li>• Cost Value c: 0.0000</li> </ul>

## Overtime Cost

Exponential cost as a function of the number of seconds of over-time for a crew's shift. The [Relative Overtime Cost](#) parameter, defined on the shift cost profile, can be used as a multiplier against this cost.

Parameter Name	Default
OVERTIME	<ul style="list-style-type: none"> <li>• Cost Value a: 2.0000</li> <li>• Cost Value b: 1800.0000</li> <li>• Cost Value c: 0.0000</li> </ul>



## Arrival Costs

Exponential cost to advance the arrival of an activity within its time window. This is calculated as a ratio of the arrival into the applicable Arrival Time Window.

- A=Multiplier. Positive values bias towards early arrival, negative towards late arrival.
- B=The ratio of the time window at which the value that A is multiplied by is doubled.
- C=Initial step value applied only once at start.

Parameter Name	Default
PKUP_SERVICE	<ul style="list-style-type: none"> <li>• Cost Value a: 1000.0000</li> <li>• Cost Value b: 1.0000</li> <li>• Cost Value c: 0.0000</li> </ul>

## Reserve Capacity Cost

Exponential cost for exceeding shift's reserve capacity. Reserve Capacity reserves a percentage of shift-time for the on-site time of activities of specified service classes. The system adds together all on-site time for non-reserved activities and all travel-time (whether for reserved activities or not). If this total exceeds the un-reserved part of the shift time, the Reserve Capacity Cost function calculates a cost from the difference.

Parameter Name	Default
RESERVE_CAPACITY	<ul style="list-style-type: none"> <li>• Cost Value a: 1.0000</li> <li>• Cost Value b: 3600.0000</li> <li>• Cost Value c: 0.0000</li> </ul>

## Shift Area Cost

Cost for the geographical size of the area covered by a shift's schedule of tasks. Promotes geographical clustering of tasks in a shift. In order to determine the size of a shift's currently scheduled area coverage, the scheduler draws a rectangle encompassing all the tasks associated with the shift and then calculates the length of the diagonal. This Shift Area Cost is applied per meter of the length of this diagonal.

Parameter Name	Default
ROUND	0.0000

## Shift Area Time Cost

Cost of time spent from the first to last task of a shift. Promotes geographical clustering of tasks in a shift.

Parameter Name	Default
ROUND_LENGTH	0.0000

## No Task In Service Area Cost

Flat cost for each shift that has no tasks within the shift's covered service areas. This cost is calculated in conjunction with the [Service Area Cost](#) parameter.

Parameter Name	Default
ROUND_ZONE	0

## Resource Attribute Cost

Cost of having a task on a shift where the task's resource-related attributes do not match those on the shift. The cost is applied once for each task with a conflicting attribute.

Parameter Name	Default
RSRC_ATTR	1,000.0000

## Depot Run Cost

Cost of activating a run.

Parameter Name	Default
RUN	1,000.0000

## Depot Run Separation

Applies a cost to loading goods during a Distribution-run.  $\text{cost} = \text{fABC} (l/c) * r$ ;

- $l$  = load added at Stop
- $c$  = capacity available on Resource
- $r$  = remaining Stops within the run

In general this parameter minimizes having load for multiple runs concurrently on board: Main example is to minimize (but not prevent) collecting return goods before the last delivery of the day has been made.

Parameter Name	Default
RUN_SEPARATION	<ul style="list-style-type: none"> <li>• Cost Value a: 10,000.0000</li> <li>• Cost Value b: 0.0200</li> <li>• Cost Value c: 0.0000</li> </ul>

## Service Class Cost

Cost of having a task on a shift where the task does not belong to any of the service classes allowed to be performed by the shift. The cost is applied once for each task with a conflicting attribute.

Parameter Name	Default
SERVICE_CLASS	1,000.0000

## Shift Promotion Cost

(separate from Shift cost)

Global cost of promoting a shift. This allows a preference to be applied for scheduling tasks to earlier shifts. The task-based Relative Shift Promotion Cost parameter, defined on activity type, can be used as a multiplier against this cost. TDC MUST be used. Default: 0, 0, 1, 48, 4, 0.042

Parameter Name	Default
SHIFT_PROMOTION	500.0000

This parameter allows you to apply variable costs to have work on a shift scheduled as close to “now” as possible. The parameter links with time dependent cost rules so that the cost changes over time. For example, you configure parameters so that the wait time for an appointment is considered as more expensive. Appointments that are scheduled immediately are less costly whereas, the further from “now” the appointment is scheduled, the more costly it becomes.

### Best Practices

- Start with 500
- If Activity Priority is not sufficiently adhered to, increase. Note that "priority" is costed. The "priority cost" is traded off against other costs like SLA and distance travelled. Therefore strict ordering by priority is not guaranteed, indeed unlikely.

## Cost for a Non-Preferred Shift Class

This cost biases the scheduler towards scheduling an activity on a shift or capacity, depending on which was specified as preferred. This only applies for organizations employing capacity based contractors.

Parameter Name	Default
SHIFTCLASS_PREFHAVE	3,000.0000

### Best Practices

- Start with 3000
- Non-Preferred Shift Class may be used to "prefer" Capacity over Shift, or vice versa. Increase the value if the scheduler is not sufficiently adhering to the preferences specified on the activities. Lower the value if "Shift Preferred" activities remain scheduled to shifts despite introducing excessive travel or overtime.

## Site Cost

Cost for non-sequential visits made to the same site. Promotes working of all activities at the same site by a single crew in a single visit.

Parameter Name	Default
SITE	0.0000

## Stop Promotion Factor

Controls prioritization of activities within a Shift.

Parameter Name	Default
STOP_PROMOTION_FACTOR	0.2000

### Best Practices

- Start with 0.2
- If priority within a Shift is not an issue, set to 0.0 to obtain the best routes.

## SLA Window Cost

Flat cost of working an activity outside its SLA time window.

Parameter Name	Default
STOP_SLA_WINDOW	0.0000

## Late Cost

Global cost of late arrival to an activity. This exponential cost is a function of the number of seconds late for arrival.

The [Relative Late Cost](#) parameter (defined at the task level) can be used as a multiplier against this cost.

Parameter Name	Default
STOPLATE	<ul style="list-style-type: none"> <li>• Cost Value a: 2.0000</li> <li>• Cost Value b: 1,800.0000</li> <li>• Cost Value c: 0.0000</li> </ul>

Late cost applied with the example allocation cost settings would drive the orders to the front of the queue globally. It is a property of the job time window.

## Travel Distance Cost

Global cost of time spent travelling, includes time contributions by site delay. The [Relative Travel Time Cost](#) parameter, defined on the shift cost profile, can be used as a multiplier against this cost.

Parameter Name	Default
TRAVELDIST	0.1000

## Travel Time Cost

Global cost of time spent travelling, includes time contributions by site delay. The [Relative Travel Time Cost](#) parameter, defined on the shift cost profile, can be used as a multiplier against this cost.

Parameter Name	Default
TRAVELTIME	1.0000

## Under Scheduled Visit Cost

Cost as a function of the number of seconds that a Complex Activity's Visit Duration is under the Target Visit Duration (Where Target Visit Duration will typically be the Average Visit Duration except for the last Visit where the Target Visit Duration will be the Fixed duration as assigned by the scheduler).

Parameter Name	Default
UNDER_SCHED_VISIT	<ul style="list-style-type: none"> <li>1.0000</li> <li>1,800.0000</li> <li>0.0000</li> </ul>

### Best Practices

- Start with 1000
- Increase if the scheduler adds too much work before and/or after Visits. First ensure the TVD is set correctly.

## Wait Time Cost

Quadratic cost for the shift being idle. The shift-based [Cost Idle](#) parameter, defined on the shift cost profile, can be used to control the cost on a shift-by-shift basis.

Parameter Name	Default
WAIT_TIME	0.0000

## Wait Time Past Cost

Quadratic cost for shift being idle in the past.

Parameter Name	Default
WAIT_TIME_PAST	0.0000

## Window Cost

Global cost of using a task's arrival time window. The [Relative Window Cost](#) parameter (defined in the task time window) can be used as a multiplier against this cost.

Parameter Name	Default
WINDOW	0.0000

## Service Area Cost

Flat cost for each task outside the shift's preferred service areas. This is used to encourage crews to work in their preferred service areas.

Parameter Name	Default
ZONE	1000.0000

## Entity Parameters

Entity parameters are generally configured on the activity type, the shift cost profile, on the task or other specific entity it relates to. The configuration location is specified in the description of each parameter.

To view these in the system, navigate to **Admin Menu > Scheduler > Parameter Definition** and select the filter for the **Entity** parameter type.

### Cost Idle

The parameter is defined on the shift cost profile.

If checked (True), the scheduler considers the cost of idle time for a specific shift's crew, weighted according to the “time from now” factor. No cost is applied if the idleness accrues further than the wait horizon time interval and, within this horizon, the applied cost is stronger if the idleness is closer to the current time.

If unchecked (False), idle time cost is not considered.

Parameter Name	Default
COST_WAIT_SHIFT	True

### Shift Distance Cost

Shift cost profile factor used as a multiplier against the global [Travel Time Cost](#). For example, a relative distance cost of 3 means that the distance cost for a particular shift is three times more than the distance cost defined globally.

Parameter Name	Default
DIST_COST	1.0000

### Relative Late Cost

Relative cost of arriving late to this activity. For example, a relative late cost of 3 means that the cost of late arrival for a particular activity is three times more than the late cost defined globally. This is used to discourage scheduling activities with planned late arrival time. This is defined at the task level.

Parameter Name	Default
LATE_COST	1.0000

### Relative Overtime Cost

Shift cost profile factor used as a multiplier against the global [Overtime Cost](#) parameter. For example, a relative overtime cost of 3 means that overtime for a particular shift is three times more than the overtime cost defined globally.

Parameter Name	Default
OVERTIME_COST	1.0000

## Relative Efficiency

Defines a mobile worker's efficiency relative to expected efficiency for mobile workers of this type. For example, if a mobile worker is twice as efficient as other mobile workers of this type, the Relative Efficiency is 2. A crew shift's relative efficiency is the average of all its allocated mobile workers' relative efficiencies.

Parameter Name	Default
REL_EFFICIENCY	1.0000

## Relative Speed

Defines a vehicle's speed relative to the expected speed for vehicles of this type. For example, if a vehicle is twice as fast as the expected speed for vehicles of its type, the Relative Speed is 2. A crew shift's relative speed is the lowest of all its allocated vehicles' relative speeds.

Parameter Name	Default
RELSPEED	1.0000

## Relative Shift Cost

Shift cost profile factor used as a multiplier against the global [Shift Cost](#) parameter. For example, a Relative Shift Cost of 3 means that the cost of activating the shift is three times more than the shift cost defined globally.

Parameter Name	Default
SHIFT_COST	1.0000

## SLA Flexibility

Period of time before and after the preferred time window (PTW) within which an activity must be scheduled to avoid incurring added cost. This is defined on the activity.

If arrival is outside this time period, the scheduler applies a fixed maximum cost (specified in the Preferred Time Factor field for the activity type). For example, if the SLA Flexibility is set to 20 minutes, then scheduled arrival must be no more than 20 minutes before the start of this preferred time window or 20 minutes after the end of this window; otherwise, the Preferred Time Factor cost is applied.

Format: hhmmss.

Parameter Name	Default
SLA_FLEXIBILITY	60.0000

## Preferred Time Factor

Fixed cost that will be applied for working an activity outside SLA Flexibility period. The cost is applied when the job is done. This is defined on the activity type and sets the cost factor for working an activity outside of the preferred time window.

Parameter Name	Default
SLA_PRIORITY	1.0000

## Relative Shift Promotion Cost

Task-based cost factor used as a multiplier against the global [Shift Promotion Cost](#). A higher value puts more pressure on the scheduler to find a cheaper (i.e., earlier) shift for tasks of this type. This value is defined at the activity type level and affects activity priority.

Parameter Name	Default
TASK_SHIFT_PROMOTION	1.0000

## Relative Travel Time Cost

Shift cost profile factor associated with travel time for a crew shift. This is used as a multiplier against the global [Travel Time Cost](#). For example, a Relative Travel Time Cost of 3 means that the cost of travel time for a particular shift is three times more than the travel time cost defined globally.

Parameter Name	Default
TIME_COST	1.0000

## Relative Window Cost

This task-based cost factor is used as a multiplier against the global [Window Cost](#). It defines the preference of an activity's time window relative to other time windows. For example, a Relative Window Cost of 3 means that the cost of using an activity's particular time window is three times more than the [Window Cost](#) defined globally.

Parameter Name	Default
WINDOW_COST	1.0000



# Addendum C

## Scheduler Parameters

This chapter provides descriptions of the administrative scheduler parameters. Cost control and entity parameters are described in [Understanding Cost Control Parameters](#). These administrative parameters should only be modified with very careful consideration and with the guidance of you Oracle representative.

### Parameter Descriptions

This section provides descriptions of the administrative scheduler parameters for reference.

Navigate to **Admin Menu > Scheduler > Parameter Definition** and select the filter for the appropriate parameter type:

Parameter Type	Description
<a href="#">Appointment Generation Parameters</a>	Controls how appointment booking requests from the host are processed.
<a href="#">Connection Parameters</a>	Controls how the scheduler sends and receives data to/from client applications.
<a href="#">Understanding Cost Control Parameters</a>	Defines how to calculate and apply costs used to build the most cost-effective schedule.
<a href="#">Entity Parameters</a>	Defines shift-based and task-based parameters that are applied against global parameters.
<a href="#">General Administration Parameters</a>	Defines system-wide scheduling parameters.
<a href="#">Logging Parameters</a>	Controls how and where the scheduler creates log files.
<a href="#">Map Configuration Parameters</a>	Defines mapping parameters used to build and optimize routes.
<a href="#">Optimizer Behavior Parameters</a>	Parameters that affect optimization of the scheduler.
<a href="#">Optimizer Performance Parameters</a>	Defines parameters that affect system performance.
<a href="#">Real Time Parameters</a>	Controls the scheduler's real-time functionality.
<a href="#">Reference Time Parameters</a>	Controls how and when reference time is updated.

Parameter Type	Description
<a href="#">Scheduler Manager Parameters</a>	Defines parameters used by scheduler manager functions.
<a href="#">Site Parameters</a>	Defines the default Coordinated Universal Time and the error translation file used to translate error messages at initialization.

## Appointment Generation Parameters

Use appointment generation parameters to configure settings for how appointments are automatically generated by the scheduler based on various criteria such as the available appointment windows, cost limits, and filtering.

### Chronological Filtering

If checked (True), non-chronological appointment windows are removed from (filtered out of) the returned list of appointment windows. If unchecked (False), non-chronological filtering is disabled.

This parameter configures the Slot Generator so that it does not return higher cost appointment slots which are earlier than later less expensive appointment slots.

Some businesses sort the appointment slots returned by date and time and tend to select the earliest slot, while disregarding the cost of the slot. This results in less efficient schedules and tends to produce unfavorable performance metrics.

Parameter Name	Default
slotGenChronologicalFilter	False

### Chronological Filtering Tolerance

Maximum number of hours an appointment window can start before the previous one. Applicable only when the chronological filtering is enabled.

This parameter configures the Slot Generator so that it does not return higher cost appointment slots which are earlier than later less expensive appointment slots.

Some businesses sort the appointment slots returned by date and time and tend to select the earliest slot, while disregarding the cost of the slot. This results in less efficient schedules and tends to produce unfavorable performance metrics.

Parameter Name	Default
slotGenChronoFilterTol	12

## Termination Threshold

Sets the number of appointment slot candidates that will be costed after the cost-limit has been reached during slot generation. During the slot generation process the slot generator may come across a group of slots that return a value greater than the MAX\_UNITS specified in the CHS\_SLOT message.

This parameter sets the number of consecutive replies above MAX\_UNITS that will be processed before the slot generator terminates.

Parameter Name	Default
slotGenTermination	30

## Window Candidate Minimum

Minimum number of candidates considered for a single time window by the appointment booking request process.

Parameter Name	Default
slotGenWinCandMin	10

## Window Filtering

Controls filtering of returned appointment windows based on time window border conditions.

**Contained within:** An appointment window must be within the request time window.

**Starts within:** An appointment window must start within the request time window.

**Ends in:** An appointment window must end within the request time window.

**Overlaps:** An appointment window must overlap with the request time window.

Note that these conditions apply on top of the initial selection criteria indicating that the task can be scheduled to be worked at some time during the appointment window, based on the availability of suitable crew and task time window.

The filters are explained in the following example:

A request was sent with 8am to 10pm time window and the slots are specified as 7-12, 14-19, 19-23.

- For filter (SG\_CONTAINED\_IN) 14-19 slot will be considered
- For filter (SG\_STARTS\_IN) 14-19 and 19-23 slots will be considered
- For filter (SG\_ENDS\_IN) 7-12, 14-19 slots will be considered
- For filter (SG\_OVERLAPS) 7-12, 14-19, 19-23 slots will be considered

Parameter Name	Default
slotGenWinFilter	SG_CONTAINED_IN

## Connection Parameters

Connection parameters control how the scheduler sends and receives data to/from client applications.

### Client Timeout

Inactivity timeout interval. After this number of seconds of no activity, the scheduler will drop the connection to a client. Set to -1 to disable timeout.

This is set internally and should not be changed.

Parameter Name	Default
clientTimeout	-1

### IO Buffer Size

Size (in kilobytes) of the data buffer used by IPC sockets for sending and receiving data. Used to set a sufficient buffer size without wasting memory resources. This is set internally and should not be changed.

Parameter Name	Default
io_bufsize	128

### Maximum Packets

Maximum number of packets that will be read in one socket poll. This should not be changed.

Parameter Name	Default
maxPackets	1000

### Maximum Wait Block

Maximum time (in seconds) to wait on a blocked socket before closing the socket.

Parameter Name	Default
maxWaitBlock	300

### Poll Time Interval

Time interval at which sockets are polled for incoming data. This value should not be changed.

Used to set the optimum rate of communication between processes.

Parameter Name	Default
pollTimeInterval	0.1

## Server Mode

If True, sets the mode of operation to Server and Client. In this mode, the scheduler immediately grants connection to all requesting clients.

If False, the scheduler first obtains a connection from the server process and only then grants connection to all requesting clients.

Parameter Name	Default
serverMode	False

## Socket Buffer Size

Internal socket buffer size for sending data. This is the maximum amount of memory (in kilobytes) that can be allocated per socket. It should be set to at least twice the size of the schedule.

This value should be set to at least twice the size of the schedule. The schedule size can be determined by saving to a '.hip' file.

Parameter Name	Default
socketBufferSize	20000

## General Administration Parameters

General administration parameters control system-wide scheduling parameters.

### Test Level

Refers to additional code for performing run-time consistency checks.

- 0: no optional tests
- 1: consistency tests that have a negligible impact on performance
- 3: consistency tests that have some impact on performance
- 5: consistency tests that may significantly alter performance

Normal setting is 0 and should not be changed in a production environment.

Parameter Name	Default
testLevel	0

## Logging Parameters

Logging parameters control how and where the scheduler creates log files.

### Auto Save Directory

Directory path name in which auto save files are stored.

Parameter Name	Default
autosaveDirectory	AutoSave

## Auto Save Files

Number of auto save files to store.

Parameter Name	Default
autosaveFiles	20

## Auto Save Interval

Interval (in minutes) between each autosave. When set to zero, auto save is disabled.

Parameter Name	Default
autosaveInterval	0

## Backup Auto Save Directory

Directory path name in which backup auto save files are stored.

Parameter Name	Default
dualAutosaveDirectory	0

## Log Compression

File compression method for log files. Compression of log files is available only for UNIX installations. The default is gzip.

Parameter Name	Default
logCompress	gzip

## Log File

Path and prefix of log files. Must be a valid and accessible location.

Parameter Name	Default
logFilePrefix_MSG	LOG\smauto

## Log Hours

Number of hours each log file spans. Log files are usually sent as e-mail attachments. Setting this value to 1 hour will keep file size to a minimum.

Parameter Name	Default
logHours_MSG	1

## Log Level

Level of detail that will be logged for the process. Log levels:

0=No logging

1=Logs real-time/operator events/status changes, shift/ task status changes, etc.

2=Logs operation of automated processes impacting schedule execution, dead-reckoning, auto dispatch, etc.

3=Operation of optimizer

4=Anything else

5=Spare

6-9=Same as levels 1-4, but accompanied by performance penalties or excessive output

Parameter Name	Default
logLevel	3

## Log to File

If checked (True), saves all log messages to a file. This should always be True.

Parameter Name	Default
logToFile_MSG	TRUE

## No Log to Standard Output

If checked (TRUE), logging to standard output is disabled.

If not checked (FALSE), logging to standard output is enabled.

Parameter Name	Default
noLogStdOut	TRUE

## Number of Log Files

Number of log files stored.

Parameter Name	Default
numLogFiles_MSG	168

## Received HIP Logging Flag

If checked (True), all the HIP packets received from the scheduler are logged.

If unchecked (False), received packets are not logged.

Parameter Name	Default
rxHipLogging	TRUE

## Memory Logging

Enables memory logging within the scheduler

If checked (True), all are logged.

If unchecked (False), memory is not logged.

Parameter Name	Default
shMallocMemLog	FALSE

## Transmitted HIP Logging Flag

If checked (True), all HIP packets sent from the scheduler are logged.

If unchecked (False), sent packets are not logged.

Parameter Name	Default
txHipLogging	TRUE

## Map Configuration Parameters

Map configuration parameters are used to build and optimize routes.

### Log Map

If checked (True), the system logs the number of map nodes, links, and matrix rows loaded.

If unchecked (False), these items are not logged.

Parameter Name	Default
logMap	FALSE

### Map File

Directory path and filename of the connectivity map for use by the scheduler. 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. This data is used to create the schedule.

Parameter Name	Default
map	0



## Matrix Distance Factor

Distance factor (integer multiplier) used in the objective function to calculate the best route. If this is set to zero, the fastest routes are calculated. If [Matrix Time Factor](#) matrixTimeFactor is set to zero, the shortest routes are calculated. The contribution of distance and time are approximately equal when the ratio of [Matrix Time Factor](#) / Matrix Distance Factor matrixTimeFactor / matrixDistFactor is approximately 20. Do not change unless advised by Oracle.

Parameter Name	Default
matrixDistFactor	1

## Matrix Segmentation

If checked (True), segmentation matrix is generated only for tasks that are compatible with the shifts.

If unchecked (False), segmentation matrix will be generated for all tasks in the system.

Parameter Name	Default
matrixSegmentation	TRUE

## Matrix Time Factor

Time factor (integer multiplier) used in the objective function to calculate the best route. If this is set to zero, the shortest routes are calculated. If [Matrix Distance Factor](#) is set to zero, the fastest routes are calculated. The contribution of distance and time are approximately equal when the ratio of Matrix Time Factor / [Matrix Distance Factor](#) matrixTimeFactor / matrixDistFactor is approximately 20. Do not change unless advised by Oracle.

Parameter Name	Default
matrixTimeFactor	40

## Matrix Toll Factor

Toll factor (integer multiplier) used in the objective function to calculate the best route. Used to discourage specific roads or road-segments.

Parameter Name	Default
matrixTollFactor	0

## Matrix Update Interval

Interval (in seconds) to check the matrix for unused cells.

Parameter Name	Default
matrixUpdateInterval	900

## Matrix Version

Matrix implementation version. Options include:

- Full matrix, no segmentation
- High connectivity
- Low connectivity
- Sparse matrix, much segmentation
- Sparse matrix, symmetrical segmentation

The default is set by Oracle for best map performance and should not be changed.

Parameter Name	Default
matrixVersion	LOCONN

## Off Map Speed

Used if the specified coordinates do not match the road-network. This is the travel speed (meters/ per second) between a task location and the nearest road or other task location.

Parameter Name	Default
offMapSpeed	20

## Route Cache Memory Limit

Maximum amount of memory the route cache will use.

Parameter Name	Default
routeCacheMemoryLimit	10

## Warning Distance

Sets the maximum distance that task locations are allowed to be from the nearest map road before a warning is issued.

Parameter Name	Default
warnDist	5000

## Optimizer Behavior Parameters

Optimizer behavior parameters affect how the scheduler creates the most efficient and effective travel routs, arrival times, routes, etc.

### Apply Minimum Travel Time to All Tasks

If checked (True), minimum travel time between tasks is applied for all tasks.

If checked (False), minimum travel time is applied between activity tasks only.

Parameter Name	Default
applyMinTravelTimeToNonStop	FALSE

## Arrival Margin

Desired number of minutes prior to the start of the time window for arriving at a task.

Parameter Name	Default
arrivalMargin	0

## Expected Average Work Rate of Visits

Used by the scheduler to estimate completion time for a complex active for which not all visits have been scheduled.

Parameter Name	Default
avgVisitsPerDayForMST	1.0

## Unexpected Event Handling

If checked (True), the system automatically identifies and resolves issues caused by unexpected events. For instance, a long duration task may be unexpectedly allocated to a shift, introducing 3 hours of overtime. The system will then identify activities that may be transferred to other shifts, in order to relieve overtime.

If unchecked (False), this feature is disabled.

Parameter Name	Default
bUEC	FALSE

## Maximum Compatibility Distance

This parameter controls scheduler behavior in terms of selecting compatible shifts for stops within this maximum distance.

Parameter Name	Default
compatDistanceMax	400

## Cost by Round

Determines how round-based costs are bounded. Set to TRUE to have round costs bounded by DepotStops. Set to FALSE to have round costs bounded by LogStops. "Round" based costs are usually calculated on a depot-based run, but may optionally be calculated on an entire Shift.

Parameter Name	Default
costByRound	TRUE

## Depot Capacity Time Window Rule

Determines which DepotTW an Activity's load is assigned to. Legacy:

earliest DepotTW that overlaps the Shift in AB;

Arrival DepoTW in Scheduling;

EarliestOverlappingShift: earliest DepotTW that overlaps the Shift;

EarliestOfDay: earliest DepotTW that starts within the Day;

Parameter Name	Default
DepotCapacityTWRule	Legacy

### Calculate Depot Late Cost by Arrival Time

Set to true to calculate the cost of being late to a depot starting from the arrival time.

Parameter Name	Default
depotLateByArrival	TRUE

### First Task Distance Factor

Multiplier of the distance cost applied to the first task on a shift. Promotes the initial assignment of activities close to a crew's starting location.

Parameter Name	Default
firstJobDistanceFactor	1

### Optimization Level

Various operations (such as allocating an activity to a shift, initial assignment of a new activity to a shift, etc.) require the sequencing of tasks on a shift to be optimized. This parameter determines how much effort is expended on this kind of optimization. Options are None, Standard, Thorough.

Parameter Name	Default
idbOptLevel	Standard

### Immediate Assignment

If checked (True), all mandatory tasks will be assigned to a shift immediately on receipt by the scheduler.

If unchecked (False), tasks will be placed on the free list for later, gradual insertion.

Parameter Name	Default
immediateAssignment	FALSE

### Load Factor

Sets the maximum size of goods for a single Activity relative to the vehicle. This is used in the Stop-Shift compatibility check.

Parameter Name	Default
loadFactor	1

## Maximum Overlap Between Visits

Maximum allowed overlap between shifts scheduled for the same complex activity, in minutes.

Parameter Name	Default
maxShiftOverlapForMST	15

## Maximum Upload Interval

Maximum period (in seconds) between improvement to a solution and its uploading to the database. This is used to ensure that the scheduler periodically saves the improvements to the current solution to the database so they are available to other processes. If set to zero, every solution improvement is saved to the database immediately.

Parameter Name	Default
maxUploadInterval	30

## Minimum Travel Time

Minimum travel time (in seconds) between tasks. The scheduler will adjust the travel time if it falls below this value. The system issues a warning if you try to set this value to greater than 600 seconds (10 minutes), as it can have a severely adverse effect on the schedule.

Parameter Name	Default
minTravelTime	0

## Number of Tasks Inserted

When there are multiple jobs on the free list, this is the maximum number that will be included in the plan at any one time when not in an Immediate Assignment mode.

Parameter Name	Default
nJobsInsert	10

## Over-Scheduling Warning Threshold (hours)

Establishes the threshold, above which an error is generated for over scheduling a complex activity by manually allocating visits..

Parameter Name	Default
overschedThreshold	4

## Plan Max Merge Distance

Sets a value to define a distance so that the plan-optimizer creates a new run when the crew travels above that duration. Entered in seconds. This is used to control Scheduler behavior in terms of breaking runs in two.

Parameter Name	Default
planMaxMergeDistance	3600

### Plan New Run Probability

Sets a value to define the minimum probability that the plan-optimizer will create a new run. This is used to control scheduler behavior in terms of breaking runs in two.

Parameter Name	Default
planNewRunProbability	0.1

### Relative Overcapacity

Depot-window capacity calculated based on relative overcapacity.

Parameter Name	Default
relOverCap	FALSE

### Absolute Round Extension

Determines the absolute rounding extension for variable shifts.

Parameter Name	Default
roundExtensionAbs	20.0

### Relative Round Extension

Determines the relative rounding extension for variable shifts.

Parameter Name	Default
roundExtensionRel	.2

### Same Site Factor

Multiplier to control allowed ratio between number of sites and number of tasks in a cluster. Do not change without consulting Oracle.

Parameter Name	Default
sameSiteRestrict	5

### Shift Window Extension

Number of minutes the scheduler can artificially extend the shift time window for the purpose of optimization when considering compatibility with the tasks. This defines the mismatch allowed between the shift window and task time windows.

If this is set to zero, the shift and task time windows must overlap for allocation to be possible.

If this is greater than zero, then a time window mismatch within this number of minutes will be acceptable to the scheduler when making recommendations.

Parameter Name	Default
ShiftWinExtension	60

### Task Late By Arrival

If checked (True), the [Late Cost](#) is applied to tasks based on comparison with arrival time.

If unchecked (False), the late cost is applied to tasks based on the predicted departure time.

Parameter Name	Default
stopLateByArrival	TRUE

### Time Windows Segmentation

Defines a boundary (in seconds) to segment time between time windows for calculating the time of execution, late/idle time and cost. A value of -1 splits time in the middle between time windows.

Parameter Name	Default
timeWindowsSegmentation	-1

### Unexpected Event Handling Threshold

Cost threshold that must be reached in order to activate the Unexpected Event Handling functionality.

Parameter Name	Default
UEC_costThreshold	1.00E+100

### Under-Scheduling Warning Threshold (hours)

The threshold, above which a warning is generated for when a complex activity is under-scheduled.

Parameter Name	Default
underschedThreshold	8 hours

### Wait Horizon

The time interval from the current time when the Wait Time Cost is applied.

Parameter Name	Default
waitHorizon	2 hours

## Optimizer Performance Parameters

Optimizer performance parameters affect how efficiently and effectively the system performs with respect to the scheduler.

### Allocation Initial Frequency

The frequency at which the allocation shuffler is activated when the scheduler starts. The default is set to 0.1 internally and should not be changed without consulting Oracle.

Parameter Name	Default
allocMove_initRelFreq	0.1

### Allocation Final Frequency

Frequency at which the allocation shuffler is activated when the scheduler finishes. The default is set to 0.1 internally and should not be changed without consulting Oracle.

Parameter Name	Default
allocMove_lastRelFreq	0.1

### Allocation Swap Initial Frequency

The frequency at which the allocation swap shuffler is activated when the scheduler starts. This shuffler combines the Allocation and DeAllocation shuffler operations, e.g. exchanges "free" Activities. This is required for visits. The default is set to 0.1 internally and should not be changed without consulting Oracle.

Parameter Name	Default
allocSwap_initRelFreq	0.1

### Allocation Swap Final Frequency

The frequency at which the allocation swap shuffler is activated when the scheduler finishes. The default is set to 0.1 internally and should not be changed without consulting Oracle.

Parameter Name	Default
allocSwap_lastRelFreq	0.1

### Chain Insert Initial Frequency

Frequency at which the chainInsert shuffler is activated when the scheduler starts. This shuffler tries to assign an entire, or sections of a chain. The default is set to 1.0 internally and should not be changed without consulting Oracle.

Parameter Name	Default
chainInsert_initRelFreq	1.0

### Chain Insert Final Frequency

Frequency at which the chainInsert shuffler is activated when the scheduler finishes. The default is set to 1.0 internally and should not be changed without consulting Oracle.

Parameter Name	Default
chainInsert_lastRelFreq	1.0

### Chain Move Initial Frequency

Frequency at which the chainMove shuffler is activated when the scheduler starts. This shuffler tries to re-assign an entire chain, or sections of a chain. The default is set to 0.1 internally and should not be changed without consulting Oracle.

Parameter Name	Default
chainMove_initRelFreq	0.1



### Chain Move Final Frequency

Frequency at which the chainMove shuffler is activated when the scheduler finishes. The default is set to 0.1 internally and should not be changed without consulting Oracle.

Parameter Name	Default
chainMove_lastRelFreq	0.1

### Cluster Initial Frequency

Frequency at which the cluster shuffler is activated when the scheduler starts. The default is set to 1.0 internally and should not be changed without consulting Oracle.

Parameter Name	Default
clusterMove_initRelFreq	1

### Cluster Final Frequency

Frequency at which the cluster shuffler is activated when the scheduler finishes. The default is set to 1.0 internally and should not be changed without consulting Oracle.

Parameter Name	Default
clusterMove_lastRelFreq	1

### Cost Limit

The scheduler is suspended when the solution cost goes below this value.

Parameter Name	Default
costLimit	0

### Deallocation Initial Frequency

Frequency at which the deallocation shuffler is activated when the scheduler starts. The default is set to 0.1 internally and should not be changed without consulting Oracle.

Parameter Name	Default
deAllocMove_initRelFreq	0.1

### Deallocation Final Frequency

Frequency at which the deallocation shuffler is activated when the scheduler finishes. The default is set to 0.1 internally and should not be changed without consulting Oracle.

Parameter Name	Default
deAllocMove_lastRelFreq	0.1

### Depot Move Initial Frequency

Frequency at which the depotMove shuffler is activated when the scheduler starts. The default is set to 0.5 internally and should not be changed without consulting Oracle.

Parameter Name	Default
depotMove_initRelFreq	0.5

### Depot Move Final Frequency

Frequency at which the depotMove shuffler is activated when the scheduler finishes. The default is set to 0.5 internally and should not be changed without consulting Oracle.

Parameter Name	Default
depotMove_lastRelFreq	0.5

### Fill Idle Time Initial Frequency

Frequency at which the fill idle shift shuffler is activated when the scheduler starts. The default is set to 0.1 internally and should not be changed without consulting Oracle.

Parameter Name	Default
fillIdle_initRelFreq	0.1

### Fill Idle Time Final Frequency

Frequency at which the fill idle shift shuffler is activated when the scheduler finishes. The default is set to 0 internally and should not be changed without consulting Oracle.

Parameter Name	Default
fillIdle_lastRelFreq	0

### Fill Shift Initial Frequency

Frequency at which the fill shift shuffler is activated when the scheduler starts. The default is set to 0.1 internally and should not be changed without consulting Oracle.

Parameter Name	Default
fillShift_initRelFreq	0.1

### Fill Shift Final Frequency

Frequency at which the fill shift shuffler is activated when the scheduler finishes. The default is set to 0.1 internally and should not be changed without consulting Oracle.

Parameter Name	Default
fillShift_lastRelFreq	0.1

### Fill Gap Initial Frequency

Frequency at which the fill gap shuffler is activated when the scheduler starts. The default is set to 0 internally and should not be changed without consulting Oracle.

Parameter Name	Default
fillTheGap_initRelFreq	0

### Fill Gap Final Frequency

Frequency at which the fill gap shuffler is activated when the scheduler finishes. The default is set to 0 internally and should not be changed without consulting Oracle.

Parameter Name	Default
fillTheGap_lastRelFreq	0

### Maximum Exponential Rate

Places a cap on the rise of exponential costs.

When the B value in exponential cost has been used this many times to double the A value, the latter will no longer be doubled.

The default value of 0 denotes true exponential costs.

Parameter Name	Default
maxTerm	3

### mstRsrcMove Initial Frequency

Frequency at which the mstRsrcMove shuffler is activated when the scheduler starts. This shuffler tries "single crew" Complex Activities on different Crews. The default is set to 0.1 internally and should not be changed without consulting Oracle.

Parameter Name	Default
mstRsrcMove_initRelFreq	0.1

### mstRsrcMove Final Frequency

Frequency at which the mstRsrcMove shuffler is activated when the scheduler's improvement rate slows down. The default is set to 0.1 internally and should not be changed without consulting Oracle.

Parameter Name	Default
mstRsrcMove_lastRelFreq	0.1

### Number of Optimization Runs

Number of distinct optimization runs that the scheduler should perform to get itself out of local minima. This applies only when [Scheduler Mode](#) is set to either Fixed Real Time or Fixed CPU Time. The default is set to 1 internally and should not be changed without consulting Oracle.

Parameter Name	Default
nOptRuns	1

## Cycle Cascade Chance

Average number of additional scheduler subcycles. Setting this to a larger number allows the scheduler to use a broader strategy when searching for cost improvements. If a particular solution appears stagnant, then some additional scheduler sub-cycles may free the restriction. This is set to 0.2 internally and should not be changed without consulting Oracle.

Parameter Name	Default
pcCascadeChance	0.2

## Relative Cluster Size

Multiplier for the number of tasks in cluster. The scheduler generally tries to improve the schedule by selecting a set of geographically close activities. This set is referred to as "cluster".

Parameter Name	Default
relClusterSize	1

## Run Move Initial Relative Frequency

Sets a value to define the initial relative frequency for the runMove shuffler.

Parameter Name	Default
runMove_initRelFreq	0.1

## Run Move Last Relative Frequency

Sets a value to define the final relative frequency for the runMove shuffler.

Parameter Name	Default
runMove_lastRelFreq	0.1

## Scheduler Duration

Number of seconds that scheduler should continue to optimize.

If the [Scheduler Mode](#) parameter is set to Variable or Forever, this value controls the temperature.

If the scheduler mode is set to Fixed Real Time or Fixed CPU Time, this value limits the optimization time to the required amount of CPU-time or wall-time. Optimization time is also limited by recalculating the temperature based on time left.

Parameter Name	Default
schedDuration	600

## Scheduler Heat Rate

Temperature increase/unit drop in cost. The resulting value can be seen in the scheduler log as "t=nnnn." If the resulting value does not rise on making large cost improvements, then scheduler heat rate may need to be increased.

Parameter Name	Default
schedHeatRate	0.5

## Scheduler Maximum Temperature

Highest scheduling temperature. Temperature used when the scheduler is started and highest temperature during normal scheduling. Higher temperatures generate better final schedules, but take longer to generate good/acceptable schedules.

Also refer to Scheduler Heat Rate.

Parameter Name	Default
schedMaxTemp	200000

## Scheduler Minimum Temperature

Lowest scheduling temperature. Temperature used when scheduling nears completion.

Parameter Name	Default
schedMinTemp	400

## Scheduler Mode

The length of time that the scheduler will continue to improve the solution before terminating.

This parameter should be set to FOREVER for seamless scheduling.

- **Forever:** Schedules into infinity, so progress is always 0.
- **Variable:** Finds the remaining amount of time. This is linearly related to the log of the relative temperature.
- **Fixed Real Time:** Scheduling will terminate at the specified wall-time.
- **Fixed CPU Time:** Scheduling will terminate at the specified CPU-time.

Parameter Name	Default
schedMode	Forever

## Segment Initial Frequency

Frequency at which the segment shuffler is activated when the scheduler starts. The default is set to 1.0 internally and should not be changed without consulting Oracle.

Parameter Name	Default
segmentMove_initRelFreq	1

## Segment Final Frequency

Frequency at which the segment shuffler is activated when the scheduler finishes. The default is set to 1.0 internally and should not be changed without consulting Oracle.

Parameter Name	Default
segmentMove_lastRelFreq	1

### Segment Swap Initial Frequency

Frequency at which the segmentSwap shuffler is activated when the scheduler starts. This shuffler exchanges route-segments of two shifts. The default is set to 0.2 internally and should not be changed without consulting Oracle.

Parameter Name	Default
segmentSwap_initRelFreq	0.2

### Segment Swap Final Frequency

Frequency at which the segmentSwap shuffler is activated when the scheduler finishes. The default is set to 0.2 internally and should not be changed without consulting Oracle.

Parameter Name	Default
segmentSwap_lastRelFreq	0.2

### Shift Initial Frequency

Frequency at which the shift shuffler is activated when the scheduler starts. The default is set to 0.1 internally and should not be changed without consulting Oracle.

Parameter Name	Default
shiftMove_initRelFreq	0.1

### Shift Final Frequency

Frequency at which the shift shuffler is activated when the scheduler finishes. The default is set to 0.1 internally and should not be changed without consulting Oracle.

Parameter Name	Default
shiftMove_lastRelFreq	0.1

### Tasks In Search Area

Number of tasks that will determine the search area. The search area limits the maximum number of compatible tasks that can be moved by a cluster at one time. The default is set to 50 internally and should not be changed without consulting Oracle.

Parameter Name	Default
stopsInSearchArea	50

### Task Grouping Travel Time

Maximum travel radius (in seconds) for a group of tasks. Used for grouping tasks, this determines how far apart tasks can be and still belong to the same group. If the tasks have overlapping time windows and lie within the specified travel radius, then the scheduler will group them for allocation to the same crew.

Parameter Name	Default
tspClassTime	0

## Real Time Parameters

Real-time parameters control how the scheduler functions in real-time.

### Auto Dispatch Maximum Load

Maximum load as a ratio of vehicle capacity for Auto Dispatch.

Parameter Name	Default
AD_MaxLoad	1.0

### Auto Dispatch Minimum Load

Minimum load as a ratio of vehicle capacity for Auto Dispatch.

Parameter Name	Default
AD_MinLoad	0.0

### Auto Dispatch Rule

In case goods are scheduled to be picked up at a Depot before the previous run has been completed, this flag controls whether the Scheduler dispatched both runs at the same time, or one by one. ONE\_AT\_A\_TIME: single run only; COMBINED: logical run;

Parameter Name	Default
AD_Rule	ONE_AT_A_TIME

### Auto Dispatch Stability Period

Minimum period of stability (in seconds) at a task location before auto dispatching. A stability period is applied to each task that has been nominated for auto dispatching. A task is considered stable if its shift or work sequence has not been changed for at least the specified stability period.

Parameter Name	Default
AD_StabilityPeriod	60

### Auto Dispatch Interval

Determines how often (in seconds) the automatic dispatching module checks for tasks to dispatch.

Parameter Name	Default
autoDirectInterval	60

### Auto Dispatch On Completion

If checked (True), the scheduler will auto dispatch the next task when the previous task is completed.

If unchecked (False), auto dispatch will occur when the previous task is started.

Parameter Name	Default
autoDirectOnComplete	FALSE

## Auto Dispatch Time Horizon

Maximum allowable idle time (in minutes) before automatically dispatching a task. This applies only when Enable Auto Dispatch is True. If the task to be dispatched next will be arrived at more than this number of minutes before it can be started, then it is auto dispatched only after the time period specified here.

Parameter Name	Default
autoDirectTimeHorizon	0

## Break Dispatch Mode

Defines the timing in which breaks are automatically dispatched.

**Dispatch all at Shift start:** All breaks on the started shift are auto dispatched regardless of the shift's drip horizon.

**Dispatch all within horizon:** Only breaks within the shift's drip horizon are auto dispatched.

Parameter Name	Default
breakDespatchMode	HORIZON

## Auto Dispatch Close Distribution Run

Close run when AutoDirect DIST stop.

TRUE stops additional work being added to the run.

FALSE allows additional work to be added to the run.

Parameter Name	Default
closeDistributionRun	TRUE

## Emergency Dispatch Mode

Defines emergency task dispatch considerations.

**Allow Onsite Crews:** An emergency task can be dispatched to a crew even if the crew is working onsite on another task.

**Disallow Onsite Crews:** An emergency task will not be dispatched to a crew that is currently onsite on another task.

Parameter Name	Default
emergency_despatch_mode	CONSIDER_ONSITE_

## Emergency Dispatch Time Left

The scheduler will not consider a crew for an emergency task if the estimated time to complete its current activity is greater than this value. If set to 0, there is no limit.

Parameter Name	Default
emergency_despatch_time	0

## Enable Auto Dispatch

If checked (True), auto dispatching of activities is enabled.



If unchecked (False), auto dispatch is disabled.

Parameter Name	Default
enableAutoDirect	TRUE

## Non-Productive Task Dispatch Mode

Defines the timing in which non-productive tasks (NPT) are automatically dispatched. Options are:

**Dispatch all on shift start:** All NPTs on the started shift must be auto dispatched regardless of the shift's drip horizon.

**Dispatch all within horizon:** NPTs are auto dispatched only if they are within the shift's drip horizon. This is the default.

Parameter Name	Default
pouDespatchMode	HORIZON

## POU Dispatch Mode

Defines the timing in which periods of unavailability (POU) are automatically dispatched. Valid modes:

**Dispatch all on shift start:** All POUs on the started shift must be auto dispatched regardless of the shift's drip horizon.

**Dispatch all within horizon:** POUs are auto dispatched only if they are within the shift's drip horizon. This is the default.

Parameter Name	Default
meetingDespatchMode	HORIZON

## Number of Tasks to be Dispatched

Indicates the number of tasks to be dispatched in horizon-dispatch at a time.

Parameter Name	Default
nMaxDispatchAtATime	1

## Process All at Site

If checked (True), all consecutive tasks at a site will be processed all at once.

If unchecked (False), each arrival for the same site task will be processed one at a time. Applies to dead-reckoning only.

Parameter Name	Default
processAllAtSit	TRUE

## Real Time Mode

Controls real time and dead-reckoning behavior.

**Schedule crews towards their destination:** No work can be planned before now and crews are dead-reckoned towards dispatched tasks. This is the default.

**Complete activities as scheduled:** Same as above, but additionally simulates crews working tasks. Tasks are automatically started and completed at the scheduled times.

**Real time off:** No simulated crew activity. Crew times and positions remain unchanged.

Parameter Name	Default
realTime	AUTO_POSITION

### Crew Location Time Interval

Time interval, in minutes, for relocating the crew by the dead-reckoning subsystem. The location of the crew is determined (in a real-time operation) by dead-reckoning and the occurrence of task and shift milestone events. This parameter also controls the frequency at which a task's schedule is updated for estimated time changes.

Parameter Name	Default
timeInterval	2

### Crew Back In Service

Controls when an out-of-service crew is assumed to be back in service.

If set to zero, the crew is automatically assumed to be back in service at the estimated time provided by the crew before going out of service.

If set to a non-zero value, the crew is assumed to be out of service until it actively reports back that it is in service.

Parameter Name	Default
vehicleTimeToActive	0

### Wait for Enroute

If checked (True), the system only dead-reckons towards tasks once they become Enroute.

If unchecked (False), the system also dead-reckons towards tasks that are in Closed or Dispatched state.

Parameter Name	Default
waitForEnroute	TRUE

## Reference Time Parameters

Reference time is the time used by the scheduler as input to cost-calculations that are sensitive to the current time. These parameters control how and when reference time is updated.

### Interval Reference Time

Interval (in hours) with which the reference time is updated. This is set internally to 2 hours and should not be changed without consulting Oracle.

Parameter Name	Default
intervalRT	2

## Start Time Reference Time

The time of day at which reference time is synchronized with wall-time. Reference time is used as input to various wall-time dependent functions, like TDC. Reference time is advanced at the interval specified in the [Interval Reference Time](#) parameter.

Parameter Name	Default
startTimeRT	0

## Scheduler Manager Parameters

Scheduler manager parameters are used to establish various base costs, time values, and other dependencies that affect how the scheduler manages your resources.

### Appointment Booking Maximum Cost

Sets a maximum so that the scheduler does not return an Appointment Booking when the cost exceeds this value.

Parameter Name	Default
appointmentBookingCost	50000

### Crew Shift Chooser Maximum Cost

Sets a maximum so that the scheduler does not return a shift when the cost of allocating the activity to the shift exceeds this value.

Parameter Name	Default
chooserMaximumCost	50000

### Schedule Manager Cleanse Interval

Sets a value, in seconds, to define the interval period at which the Schedule Manager checks to remove old data from the scheduler.

Parameter Name	Default
cleanseInterval	300

### Conditional Booking Maximum Cost

Sets a maximum so that the scheduler does not return a Conditional Booking when the cost exceeds this value.

Parameter Name	Default
conditionalBookingCost	50000

## Force Use of Site Functionality

Set to TRUE or FALSE to indicate whether or not site grouping should be applied to activities. If set to FALSE, site grouping only applies when a site delay is included on the activity. If set to TRUE, then site grouping applies even if there is no site delay on the activity. Refer to [Site Management](#) for more information.

Parameter Name	Default
forceSiteId	FALSE

## Shift Dependency Offset

Sets a value to specify the duration between shifts that causes them to be considered as linked. When this parameter is set to -1 shifts are never considered linked, even if their time windows overlap.

Parameter Name	Default
shiftDependOffset	3600

## Site Parameters

Site parameters define the default coordinated universal time and the error translation file used to translate error messages at initialization.

### Error Translation File

Directory path and filename of the error translation file used to translate error message text at initialization.

Parameter Name	Default
errorTranslationFile	0

### Default UTC Offset

Default UTC offset for packet field, in the format: (-)HHMM or (-)hhmmss. The default is the application time zone. The scheduler only deals with absolute, not abstract, times. Use this to pass an optional UTC offset when exporting time data.

Parameter Name	Default
hipDefaultUTCOffset	SystemTZ