

252336

Aptean Ltd
Copyright © 2011-2026.

Contents

| | |
|---|-----------|
| 1 252336..... | 1 |
| 2 252336 PA-7DGNK2 Change Schedule Date of a Scheduled Trip..... | 2 |
| 3 FUNCTIONAL OVERVIEW..... | 3 |
| 3.1 Client Requirement..... | 3 |
| 3.2 Solution..... | 3 |
| 3.3 Scope..... | 4 |
| 4 SET-UP..... | 5 |
| 4.1 Menu Structure..... | 5 |
| 4.2 Data..... | 5 |
| 5 FUNCTIONAL DESCRIPTION..... | 7 |
| 5.1 OK Command Button..... | 7 |
| 5.2 Cancel Command Button..... | 7 |
| 6 References..... | 8 |
| 7 Document History..... | 9 |
| 8 Authorised By..... | 10 |

1 252336

Ready for **What's Next, Now™**



2 252336 PA-7DGNK2 Change Schedule Date of a Scheduled Trip

Copyright OBS Logistics © 2008

The information contained herein is the property of OBS Logistics and is supplied without liability for errors or omissions. No part may be reproduced or used except as authorised by contract or other written permission. The copyright and foregoing restriction on reproduction and use extend to all media in which the information may be embodied



3 FUNCTIONAL OVERVIEW

3.1 Client Requirement

For Trips in Statuses PLANNED, TENDERED and ACCEPTED To be able to retain a scheduled Trip ID with associated orders and transfer as one to an alternative schedule date if required after the trip has been built and a loading schedule issued for the original date. Invoke change by 'Right-Click' Drop-down box [Change Schedule Date] When the Trip schedule date is changed automatically update the Order Scheduled delivery date for all orders on the Trip. Delete any resources added into the Trip on the original schedule date and revert CARRIER to OWNING DEPOT on new schedule date

3.2 Solution

Changing the Schedule name (SCHED_NAME) of a specific trip can be done, but due to referential integrity (links between data) we will have to follow specific steps to complete the change. Several tables have keys which are based on a combination of the SCHED_NAME and TRIP_ID fields. In these tables we cannot simply update SCHED_NAME as this would violate existing constraints and the Oracle Database would not allow the change.

To work around this we must make a copy of the record(s) we want to change. In the copy record we then update the SCHED_NAME. Once the record has been created and updated, we can remove the original record.

To ensure that we comply with all the data constraints we must copy and update the records from the top (SCH_TRIP table) down and then delete the original records from the bottom (SCH_TRIP_STOP, SCH_TRIP_AUDIT tables) up to SCH_TRIP table.

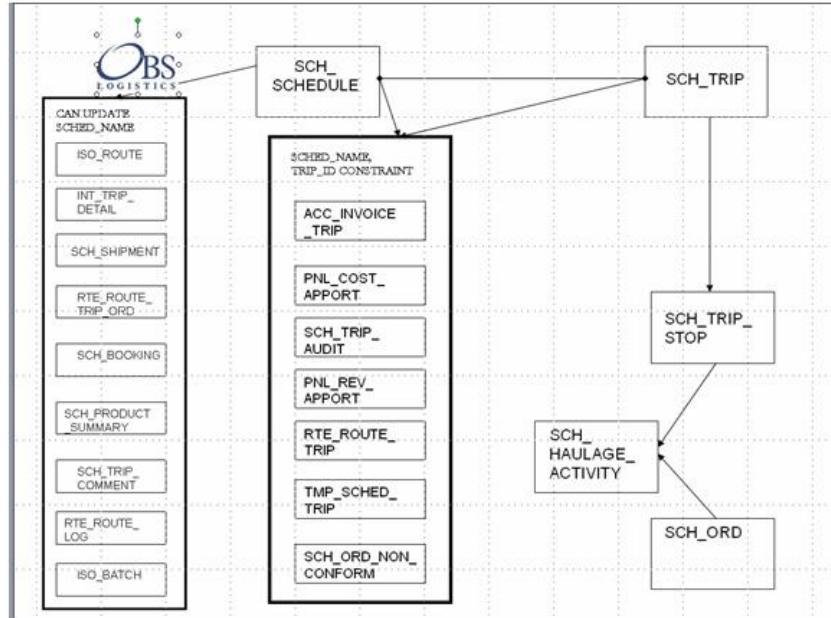
Several tables, where SCHED_NAME is a simple foreign key, the SCHED_NAME can be updated without requiring copies and deletions.

To allow you to understand the scope of this change, the referential integrity issues will effect 9 tables, with another 11 tables requiring some manipulation and updating. (please see diagram attached)

In addition to the record changes above, we need to identify which orders have been assigned to the trip using the SCH_HAULAGE_ACTIVITY table. We then update the SCHED_NAME in the SCH_ORD table for all the identified orders and update the late delivery field to match the new schedule.

To update the late delivery field, we intend to amend the date part only, to match the new SCHED_NAME and retain the time from the original schedule. Set Driver_id, trailer_id and tractor_id to Null in the old record in SCH_TRIP table and set Trailer_id to Null in the new record in the SCH_TRIP_STOP table. Update Carrier_ID on new SCH_TRIP record to equal the OWNING_DEPT. These changes will all be completed within a new Package on the form and will be launched from the Trip Tree menu. A new menu option will be added called - ?Change Schedule date?. Selecting this option will launch the new package.





3.3 Scope

This change will be applied to system version 10.5.0 on CONTST and once approved CONPRD.



4 SET-UP

4.1 Menu Structure

Pop up menu Trip Tree Menu will be amended to include a new menu item called ?Change Schedule Date? When this is launched, the status of the existing scheduled trip will be checked. If the status is not ?PLANNED?, ?TENDERED? or ?ACCEPTED? the user will be told they cannot change the date. If the trip is the required status, a small input form will launch where the user can enter the new schedule date.

4.2 Data

The new schedule date will be stored in a non-database item and used as reference for amending the relevant records. The schedule date field , on the new screen will include a list of values, which will be populated based on the schedules available in the database. The user must select a value from this list.

4.2.1 Copying data

In the following tables, we will create records based on the new schedule date, where all other values are the same for the trip that is being amended.

```
Insert into SCH_TRIP (sched_name, trip_id) VALUES (:change_schedule.new_schedule, :trip_dtl.trip_id)
```

Commit;

```
Update SCH_TRIP SET (all other fields) = SELECT (all other fields FROM SCH_TRIP where sched_name =:trip_dtl.sched_name and trip_id = :trip_dtl.trip_id) where sched_name =:change_schedule.new_schedule and trip_id = :trip_dtl.trip_id;
```

Commit;

There are now two records on the table for the same trip, one referencing the original Schedule name and one referencing the new schedule name. We must continue this process for all tables which reference the schedule_name as part of its primary key.

Before we can carry out the above steps we must first use cursors to check if the selected trip/ schedule have any related records in each table. Only if we find a related record do we insert and update. Otherwise we do nothing with the table.

4.2.2 Updating Data

Having identified the rest of the tables which reference the schedule name, we must update the relevant record(s) to reference the new schedule_name.

The order records associated with the selected trip must be updated to reflect the new schedule date and a new late delivery time. The order table does not include the trip id, so identifying the relevant orders will require linking three tables, SCH_TRIP_STOP, SCH_HAULAGE_ACTIVITY, SCH_ORD. Once the relevant records in SCH_ORD have been identified, validation will be required to ensure that the amended LATE DELIVERY DATE meets all the existing checks. We are not intending to make any changes to the late delivery time, this will simply inherit the existing value.

The last updates to be completed will remove all resources from the old trip and amending the carrier to be the owning company. In this instance, resources refers to drivers, tractors and trailers. As the old records are to be deleted, this will in fact remove the resources from the original trip.



4.2.3 Deleting Data

In several tables, there are now two trip records, each referencing a different schedule name . This will be confusing as users will not be aware which is the relevant trip. For this reason , all duplicate trip records , where the old schedule is referenced will be deleted.

Just as when we inserted data, we did so in a specific order, deleting data must also be carried out in a specific order, starting with the following tables

ACC_INVOICE_TRIP PNL_COST_APPOINT SCH_TRIP_AUDIT PNL_REV_APPOINT RTE_ROUTE_TRIP
TMP_SCHED_TRIP SCH_ORD_NON_CONFORM SCH_TRIP_STOP

And ending with the SCH_TRIP table.



5 FUNCTIONAL DESCRIPTION

This change will allow the user to select a trip, enter a different schedule date and move all the related trip information to the new schedule date. This process will be completed in three steps.

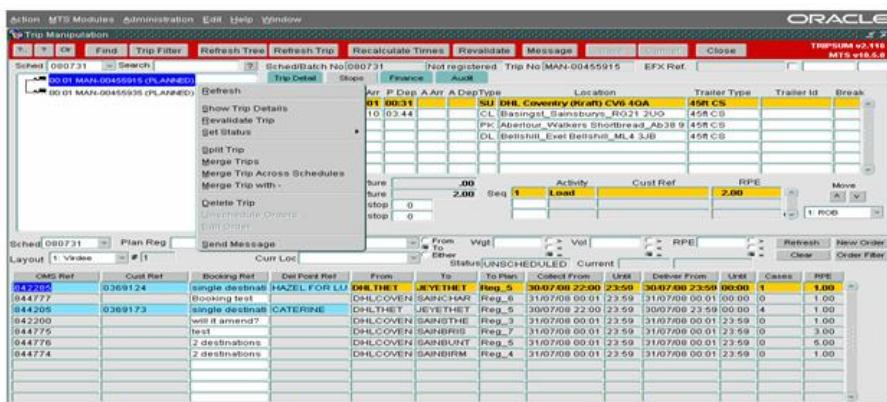
1. The user selects the trip from the trip tree and from the right click menu selects the option ?Change Schedule Date?. If the trip is in the correct status, an input form is launched.
2. The user selects the new schedule date from the list of schedule dates available. For a schedule date to be available in the list, it must exist in the SCH_SCHEDULE table.
3. The user selects OK . In the background a package is launched which carries out all the required data changes. As the final change is deleting the record that is currently on display in the form, this is an event the OK command button must handle.

5.1 OK Command Button

After carrying out all the data changes as described in the previous Data section, this button must handle the event where the record that is currently being displayed on the form is deleted. This will be achieved by clearing the data block, then setting the schedule name to be the new schedule and executing a query. This code will run after all the data changes have been committed.

5.2 Cancel Command Button

This will allow the user to cancel the schedule date change, but only before they select OK. Once OK has been selected, this cannot be cancelled. Users would be able to use the ?Change Schedule Date? functionality to revert the trip back to its original schedule date.



The above screen shot shows a trip selected in the trip tree and the right click function which launches a pop up menu. This is where the option to ?Change Schedule Date? will be available.



6 References

| Ref No | Document Title & ID | Version | Date |
|--------|---|---------|------------|
| 1 | 252336 PA-7DGNK2 change Sched date of a Sched Trip\RIO - Work Request.mht | N/A | 15/07/2008 |
| 2 | EST 252336 PA-7DGNNK2 Change Sched Date of a Sched Trip.doc | 1 | 18/08/2008 |



7 Document History

| Version | Date | Status | Reason | Initials |
|---------|----------|--------|---------------------|----------|
| 1a | 20/08/08 | Draft | Initial version | SW |
| 1 | 26/08/08 | Issue | Reviewed and Issued | JAT/RM |



8 Authorised By

| | |
|---------------------------|---------------------------|
| Dave Meir Crisford | Development Manager |
| Suk Sandhu | TMSCC MTS Product Manager |

