CTMS Paragon Interface

Aptean Ltd

Contents

1 CTMS Paragon Interface

The basic Paragon interface has been documented here: ctms:Paragon Interface and is included below by reference

This guide provides more technical information

2 Paragon Functional Guide

Note: This guide covers the direct Paragon API interface. There are other more manual interfaces to Aptean Routing & Scheduling - Paragon edition, but these are not covered here.

There are 2 types of direct Paragon APIs:

- Strategic - Fixed Drop Scheduling Engine
- Tactical - creating and optimising routes before or on the day of execution

These can be configured separately, so that either or both can be in use.

2.1 Contents

2.2 Configuration

2.2.1 System Parameters

System parameters enable the functionality of the Paragon API.

Name	Description	Usage
PAR_TENNANT_KEY1	Tennant KEY 1 for paragon API	SYSTEM
PAR_ENDPOINT_URL	URL for paragon API	SYSTEM
PAR_TENNANT	Tennant for paragon API	SYSTEM
PAR_FREQUENCY	Frequency for PAR Master keys	SYSTEM
PAR_START_DATE	Start Date for Master Keys	SYSTEM
PAR_KEY_FORMAT	PAR Master key format WKXX, DDMM, DAYX	SYSTEM
TK PAR_USE_RPOXY	Paragon Use Proxy	SYSTEM
TK PAR_PROXY	Paragon Proxy Server	SYSTEM
PAR_GROUP_STAGING	Paragon Group Staging Level	SYSTEM
PAR_RUN_NUMBER	Is Paragon Planning based on Run Numbers?	SYSTEM
PAR_KEY_PROJECT	PAR Project name	SYSTEM
PAR_API	Create Control records for Paragon API	SYSTEM
PAR_SEND_ALL_LOCATIONS	Are locations sent out via API ('TACTICAL','STRATEGIC','BOTH')	SYSTEM
PAR_AUDIT	Include auditing of the import process in the STP version of the Paragon API (Y/N)	SYSTEM
HTTPS_WALLET_FILE		
HTTPS_PASSWORD		
AUTO_SCHED_INACTIVE_DEPOTS		
TRM_RETAIN_EMPTY_STOPS		

A full list of configurable parameters is available here:

- System Parameters List

2.2.2 Order and Location Details sent to Paragon

The content of each message sent to Paragon is controlled through internal configuration tables. These are maintained and configured by your Aptean implementation team.

These allow configuration of the various elements that are sent from CTMS to Aptean Routing and Scheduling - Paragon Edition.

Orders

- Any direct field from tables:
  - SCH_ORD - the order.
  - SCH_ORDER_LINE - the deliverable types such as Parcels, Tyres, etc.
  - SCH_ORD_ITEMS - the individual parcels, or quantity of each specific product.
  - GEO_LOCATION GEO_TO - details of the final destination.
  - GEO_LOCATION GEO_FROM - details of the origin.
- Functions can be called for other information:
  - DP_PAR_API_STP.GET_REF - retrieve any reference against the order.
  - DP_PAR_API_STP.GET_ORDER_TYPE - retrieve the order type.
  - DP_PAR_API_STP.GET_FROM_LOC - summarised details of the origin.
  - DP_PAR_API_STP.GET_TO_LOC - summarised details of the destination.
  - DP_PAR_API_STP.OPENING_TIMES - opening times of the destination.
  - DP_PAR_API_STP.CLOSING_TIMES - closing times of the destination.
  - DP_PAR_API_STP.GET_DROP_NUMBER - the specific drop number.
  - DP_PAR_API_STP.GET_TOTAL_TYRES - the total tyres (specific to tyre delivery - use the below function for more generic systems)
  - DP_PAR_API_STP.GET_QTY_BY_DU - the total quantity of a specific deliverable unit, e.g. pallets, parcels, tyres, etc.

Locations:

- Any direct field from the following tables:
  - GEO_LOCATION - details of the supplied location.

2.2.3 Run Key Configuration

Aptean Routing and Scheduling - Paragon Edition controls all planning through Runs. Runs are normally associated to a schedule within CTMS, but not always. In this case, there is a Run configuration that aligns the dates of jobs within CTMS to the appropriate Paragon run key.

This is achieved through the Business Data Maintenance screen, on the *Paragon Keys* tab.

Note that the enabled in Access Control, accessible tabs, for screen "BDM" tab "PAR_KEYS".

A full list of configurable tabs and functions is available here:

- Access Control - Accessible Functionality

2.2.4 Turning on the interface

The individual processes for Paragon are controlled through EDI Process Configuration in the EDI Maintenance screen.

Inbound

Inbound processes are split into 3

- Inbound Tactical Receive per depot
- Inbound Strategic Receive per depot
- Inbound Processing for all staged receipts above

Regardless of the components of the interface that are in use, the latter process must always be running.

Inbound Strategic Route

These processes get the information from Paragon and stage the information on inbound tables, ready for import

This can be configured for all depots or one per depot, depending on how Paragon is configured. For example, if Paragon is configured with different plans per regional depot, then each import process should be configured separately for each depot here. Therefore this should be named appropriately e.g. the name of the depot.

Process: DP_PAR_API_STP.get_paragon_route

Parameters

- DEPOT_KEY - the RDC Location ID

Report Values

- Package PROCESS DP_PAR_API_STP.get_paragon_route
- Process p_process_name the name of the EDI process that has been configured above.

Inbound Tactical

These processes get the information from Paragon and stage the information on inbound tables, ready for import

This can be configured for all depots or one per depot, depending on how Paragon is configured. For example, if Paragon is configured with different plans per regional depot, then each import process should be configured separately for each depot here. Therefore this should be named appropriately e.g. the name of the depot.

Process: DP_PAR_AP_STPI.get_paragon

Parameters

- DEPOT - the EDI Process Name

Report Values

- Package PROCESS DP_PAR_API_STP.get_paragon
- Process p_process_name Get_Paragon

Inbound Processing

This is the general inbound processing job.

This process processes the information from the inbound tables into the CTMS database.

- Name: paragon_in
- PROCESS: DP_PAR_API_STP.READ_PARAGON_IN

Report Values

- Package PROCESS DP_PAR_API_STP.READ_PARAGON_IN
- Process p_process_name paragon_in

Outbound

- Name: Paragon_Outbound
- Process: DP_PAR_API_STP.process_paragon

Parameters

- AUDIT_WS Y/N
- USE_RUN_DEPOT Y/N

Report Values

- Package PROCESS DP_PAR_API_STP.process_paragon_outbound
- Process p_process_name Paragon_Outbound

2.3 Strategic Interface

This interface allows definition of locations onto fixed routes at specific drop numbers.

Note: This is applicable to Fixed Drop Scheduling engine only.

When imported, this deletes any previous configuration against locations and replaces it with the new network map.

Note: Bank Holiday routes will NOT be deleted - these are expected to be managed manually in CTMS.

The data that is sent is configurable, as seen in the sections above.

2.4 Tactical

When orders are received into CTMS, they may be planned on temporary trips using the scheduling engine - these trips should be configured to be prefixed with "TMP" so that they can be easily distinguished.

Orders and Locations are sent to Paragon for planning.

The data that is sent is configurable, as seen in the sections above.

Paragon users then optimise and plans the orders.

When these Paragon routes are frozen, these are exported back to CTMS automatically. This remove any TMP trips, creates RTE trips and sets them to TENDERED status.

3 Technical Information

3.1 Packages

- DP_PAR_API
- DP_PAR_API_STS

3.2 CONFIGURATION - OUTBOUND

```
-- EDI Interface - FROM EDI_IF_CONTROL EXTERNAL_SYSTEM LIKE 'PAR_API%'

SELECT * FROM EDI_PROCESS_HEADER
--WHERE FTP_USERNAME IS NOT NULL
WHERE PROCESS_NAME LIKE '%PAR%'

-- EDI Process Parameters
SELECT * FROM EDI_PROCESS_TRIGGERS
WHERE PROCESS_NAME LIKE '%PAR%'

-- Records are triggered to be written to Paragon from
-- TRG_GEO_LOCATION_UID - note that this does not use system parameter PAR_API
-- TRG_LOCATIONS_UID
-- TI_SCH_ORD_STATUS
-- TRG_SCH_ORDER_LINE_PAR
-- TRG_SCH_ORD_AUDIT_LOCS
-- TRG_SCH_ORD_CURRENT_DEPOT
-- TRG_SHA_TRUNK_TRIP
--Written to INT_XML_CONTROL - check interface type and order

-- NOTE: The FIX interface (for adding locations to a routing plan, to generate routes/drop-numbers)
-- is no longer in use on STSTPRD - commented out code, but still in TMSDEV - watch out for that
-- They manually add the addresses to Paragon now.

-- Packages that reference PAR_API
-- DP_CTMS_IMPORT
-- DP_FLEXIPOD
-- DP_FLEXIPOD_2
-- DP_PAR_API
-- DP_PAR_API_STP
-- DP_SCHEDULING_ENGINE_STAP
-- DP_SCHEDULING_ENGINE
-- TRM

-- Packages that reference PAR_API parameter
-- TI_SCH_ORD_STATUS
-- TRG_LOCATIONS_UID
-- TRG_SCH_ORDER_LINE_PAR
-- TRG_SCH_ORD_AUDIT_LOCS
-- TRG_SCH_ORD_CURRENT_DEPOT
-- TRG_SCH_ORD_REFERENCE_RUN
-- TRG_SHA_TRUNK_TRIP
-- TRM

-- Packages that write 'LOC' paramgon records (FIX interface)
-- DP_PAR_API_STP
-- TI_SCH_ORD_STATUS
-- TRG_GEO_LOCATION_UID

select * from int_xml_control
where external_system = 'PAR_API'
and event_type = 'ORD'
and OMS_REF = 846511
order by int_xml_seq desc

select * from int_xml_control
where external_system = 'PAR_API'
and event_type = 'LOC'
and OMS_REF = 846511
order by int_xml_seq desc
```

3.3 Parameters

Parameters - PAR_API Also

- PAR_ENDPOINT_URL
- PAR_TENNANT keys
- KEY_FORMAT

```
-- Searching for parameters
SELECT '|'||PARAM_NAME, '||'||DESCRIPTION, '||'||CONFIG_BY
FROM ADM_SYSTEM_PARAM
--SELECT * FROM ADM_SYSTEM_PARAM
WHERE PARAM_NAME LIKE '%PAR_API%'

SELECT '|'||PARAM_NAME, '||'||DESCRIPTION, '||'||CONFIG_BY
FROM ADM_SYSTEM_PARAM
WHERE PARAM_NAME LIKE 'PAR%'
FOR UPDATE
```

3.4 Tenant Keys

Keys matching dates/schedules to Paragon run keys

```
SELECT * FROM PAR_KEYS

SELECT * FROM PAR_KEYS_ROUTE
```

3.5 Outbound

```
-- Configuration of outbound processing
-- Orders to paragon
SELECT * FROM PAR_CALL_API
-- locations to paragon
SELECT * FROM PAR_CUST_API
-- Fixed routes to paragon
SELECT * FROM PAR_FIX_API


-- PICKED UP BY (EITHER/OR)
-- DP_PAR_API
-- DP_PAR_API_STS

-- PROCESS_PARAGON

-- That sends the details to Paragon.
```

3.6 Inbound

```
-- WARNING: You only want to get as few records as possible here - the results are massive.
-- So, use the following to identify a record you want using PAR_ID, the select * where PAR_ID = ?

-- Inbound Tactical
SELECT PAR_ID, PAR_DATE, PAR_PROCESSED, DEPOT_PLAN
FROM PAR_JSON_IN
ORDER BY PAR_DATE DESC

-- Inbound Strategic
SELECT PAR_ID, PAR_DATE, PAR_PROCESSED, RUN_NUMBER
FROM PAR_JSON_IN_ROUTE
ORDER BY PAR_DATE DESC

-- Where values extracted from JSON are stored for onward processing
-- Written one per order or record
select * from PAR_API_STAGING
select * from PAR_API_STAGING_ROUTE

-- INBOUND STRATEGIC
-- When staged, writes 1 record to INT_XML_CONTROL
```

```
select * from int_xml_control
where external_system = 'PARAGON_IN'
order by int_xml_seq desc

-- Typically runs DP_PAR_API_STP.READ_PARAGON_IN passing the process name as a named parameter

-- This process will ONLY process orders where the FROZEN flag is not 0
-- It does do SOME processing against non-frozen orders, but not a lot

-- When complete processing a record (frozen or not) the record is removed from staging.

-- For frozen orders, there is no auditing.
-- For non-frozen orders, there is auditing against the orders and the trips affected.

-- Other Auditing - only audits failures not successes.
SELECT * FROM EPOD_WEB_SERVICE_AUDIT
WHERE TRIP_ID = 'PARAGON'
AND EVENT_TYPE = 'FAIL'


-- INBOUND ROUTE (FIX) Interface
-- Calls Stage_paragon_route which stages to PAR_API_STAGING_ROUTE

-- Then creates an INT_XML_CONTROL record for PARAGON_IN as above (no idea why)

-- Then directly processes the records on PAR_API_STAGING_ROUTE in PROCESS_ROUTE_DATA
-- not triggered by a background process, so no ides why there is a control record created. No comments to
-- Simply creates Fixed route and stop, created from the PAR_FIX interface
select * from geo_route_dtls

-- Audits:
SELECT * FROM EPOD_WEB_SERVICE_AUDIT
WHERE TRIP_ID = 'PAR_ROUTE';
```