

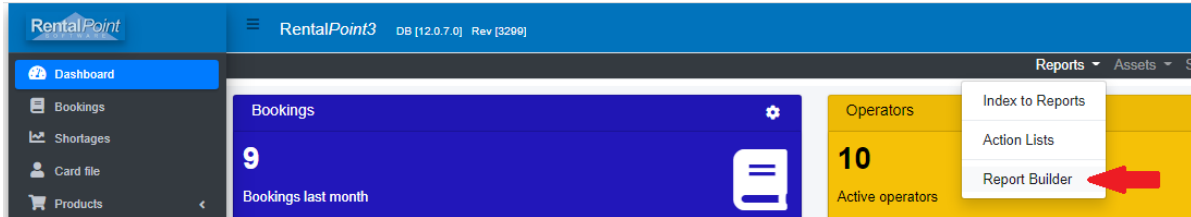
Reports & Templates



Last Modified on 25/05/2026 12:12 pm EDT

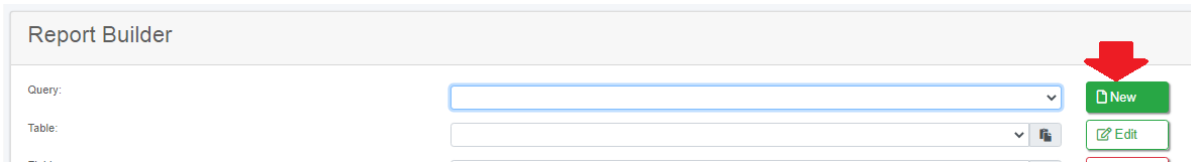
Report Builder

Occasionally you may find the need to retrieve information from RentalPoint that is not provided in standard reports. Using the Report Builder you can write an SQL query to retrieve information from the database, and save/print the output report.


Access the Report Builder from the Reports Menu

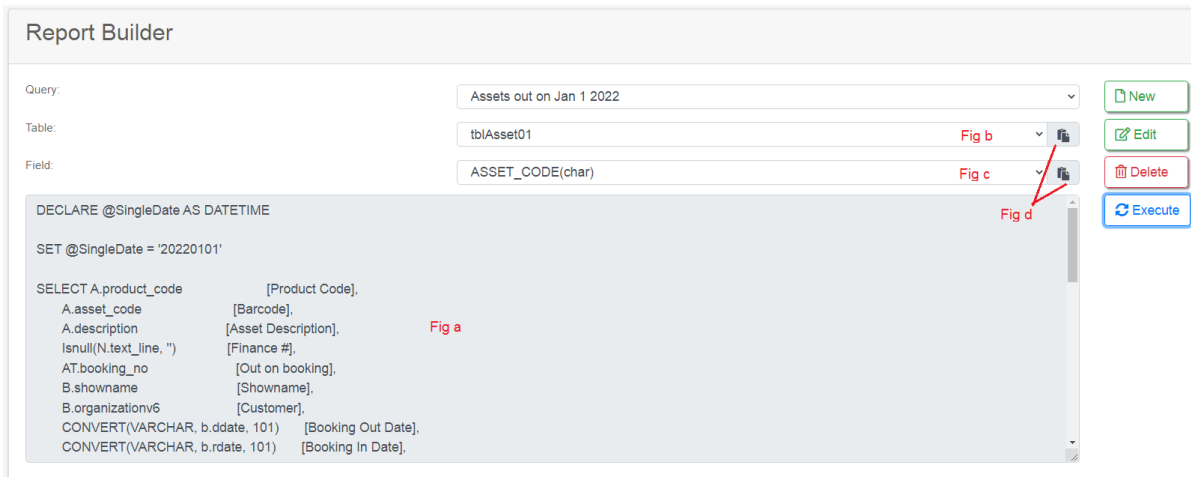




Click  to create a new report or  to edit an existing one





Once you've given your report a name, you can copy/paste an existing SQL Query into the report window, then save and execute. You'll find [some existing SQL queries in our KB](#)

Alternatively, you can build your report syntax in the syntax window (fig a below) using the drop down table list (fig b below) and the corresponding drop down field list (fig c below) to add field and table names to your syntax as needed. The  icon (fig d below) can be used to copy the selected table/field name to your current cursor position in the syntax window.



Once you  the report, you'll have an option to save  to many different formats including Excel, Word,

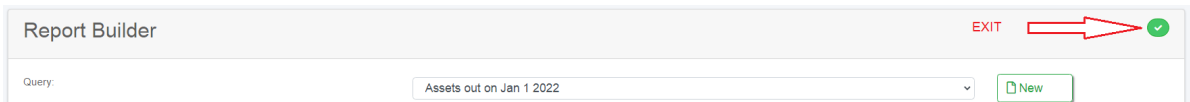
PowerPoint, txt, csv etc. and/or print  from browser or Adobe Reader. Use the  option to zoom/change the size of the output display

Close

Data from: Assets out on Jan 1 2022

Product Code	Barcode	Asset Description	Finance #	Out on booking	Showname	Customer	Booking Out Date	Booking In Date	Asset Out Date	Asset Return Date
ALTS315	ALTS315-1	Alto TS315 Speaker		SOUNDB00001	SB Fall Fest 2021	SoundByte Experience	10/31/2021	10/31/2021	12/15/2021	Still Out
ALTS315	TS3151002	Alto TS315 Speaker		SOUNDB00004		SoundByte Experience	10/05/2021	10/06/2021	10/12/2021	Still Out
ALTS315	TS3151004	Alto TS315 Speaker		JENGAR00003	St. Jacobs School Concert #1	Jennie Garth	10/14/2021	10/21/2021	10/06/2021	01/28/2022
MXPRO	12345	MX Processor		FHINK00005	Garden Party	Francis Hinkley	10/28/2021	10/28/2021	10/28/2021	02/01/2022
RCFNX15	RF1003	RF 15 Floor Standing Speaker		FHINK00005	Garden Party	Francis Hinkley	10/28/2021	10/28/2021	09/03/2021	02/01/2022
RF123	RF1231001	RF 15 Floor Standing Speaker		MGRD00007		Maria Gordon	10/06/2021	10/13/2021	10/21/2021	01/28/2022
RF123	RF1231002	RF 15 Floor Standing Speaker		FRNB00009		FRNB Events	11/04/2021	11/05/2021	11/04/2021	01/28/2022
SAMIND	SAMIND1001	60" Samsung LED Touchscreen		SOUNDB00011		SoundByte Experience	01/24/2022	01/28/2022	10/26/2021	02/01/2022
TS315CAS	TS315C1004	Alto TS315 Speaker CASE		SOUNDB00001	SB Fall Fest 2021	SoundByte Experience	10/31/2021	10/31/2021	12/15/2021	Still Out

Exit the Report Building using the  icon or simply close the window

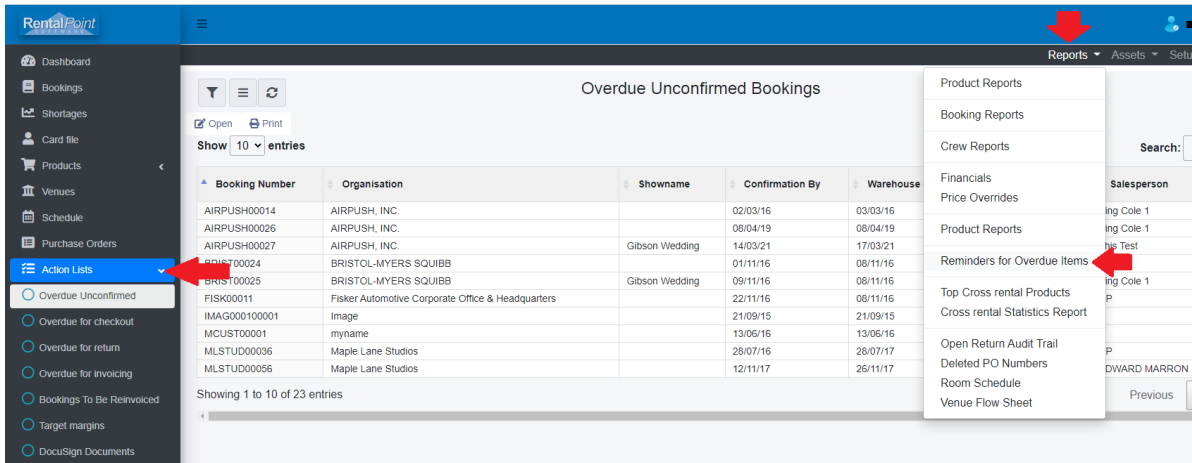


Action Lists/Reminder for Overdue Items

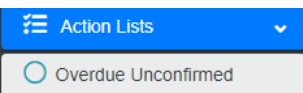

The Reminders For Overdue Items window combines several reports into one window to easily maintain all bookings overdue for specific operations, such as checkout/return/invoicing etc.

Running the Reports


Accessible from the main **Reports** menu or via **Action Lists** on the left hand menu pain



Display Preferences/Scope

Access scope for the currently highlighted report  via the  icon.

The report scope differs for each report. Use this scope area to narrow down the results displayed in the grid (eg, report for a specific location or all locations, specify a date range for some reports).

Checking and unchecking the report boxes under Display Preferences will display and hide the report tabs visible under  Your preferences will be saved for the next time you open the reminder window.

Overdue Filters FOR Unconfirmed Bookings
✕

Display Preferences

- Overdue Unconfirmed
- Overdue for Checkout
- Overdue for Return
- Overdue for Invoicing
- Bookings To Be Re-Invoiced
- Target Margins
- DocuSign documents

Region/Location

All

Select Location

8 - NY Courtyard Newark Downtown
▼

Operator Scope

All

Operator

Salesperson Scope

All

Salesperson


Status

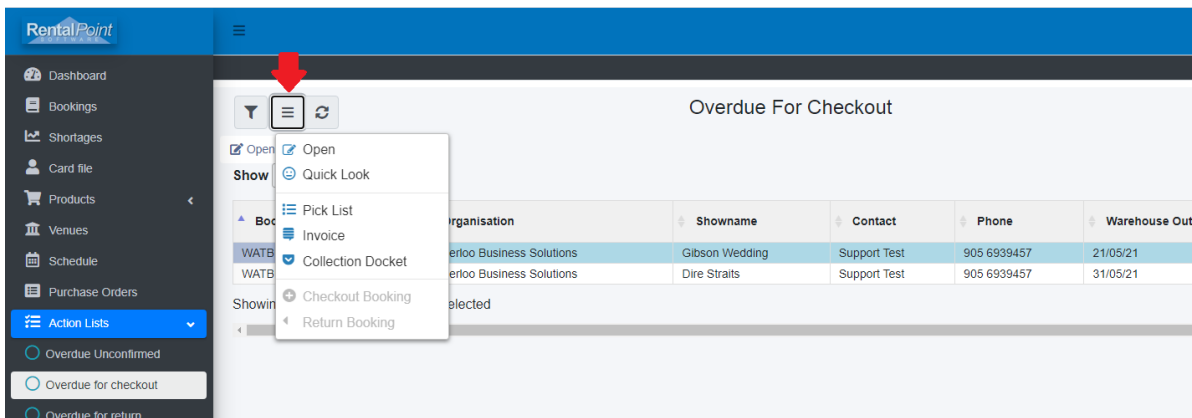
Rental

Sale

Q Ok

Interactive Operations

Each report is interactive and has several operation options such as opening a booking, viewing the audit trail and printing custom documents. Access booking operations available for the currently highlighted report via the  icon



Organisation	Showname	Contact	Phone	Warehouse Out
erloo Business Solutions	Gibson Wedding	Support Test	905 6939457	21/05/21
erloo Business Solutions	Dire Straits	Support Test	905 6939457	31/05/21

Available Reports

Overdue Unconfirmed Bookings

Lists all bookings that have not been saved in Confirmed status by the specified Confirm Date in the Status tab of the booking. Users can double click to open the booking and set it to Confirmed status, or they can use the Confirm Booking button to update the status in one click. The report automatically refreshes and removes the confirmed booking.

Overdue for Checkout

Lists all bookings that are still in Booked status and have passed their scheduled Warehouse Out date. Users can double click the booking to go directly to the checkout window. Once the checkout is completed and the booking status is set to Out, the booking will be removed from the report.

Overdue for Return

Lists all bookings that are still in Out status and have passed their scheduled Warehouse In date. Users can double click the booking to go directly to the return window. Once the return is completed and the booking status is set to Returned, the booking

will be removed from the report.

Overdue for Invoicing

Lists all bookings in confirmed status that have not been invoiced yet. Scope options allow you to narrow down the bookings you want to view. Users can double click a booking to go directly to the invoicing window.

Overdue for Project Invoicing

Lists all projects that have not been invoiced yet. Scope options allow you to narrow down the projects you want to view. Users can double click a project to go directly to the project invoicing window.

Bookings to be Re-invoiced

Lists all bookings that have been previously invoiced where the current total amount of the booking does not match the original invoiced amount. Users can double click the booking to access the invoicing window and re-invoice the booking.

Long Term to be Invoiced

Lists all long term bookings that need to be invoiced. Scope options allow you to narrow down the bookings you want to view (ie. 7 day, 14 day, monthly etc). Users can double click a booking to go directly to the invoicing window.

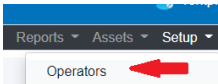
Overbooked Cross Rentals

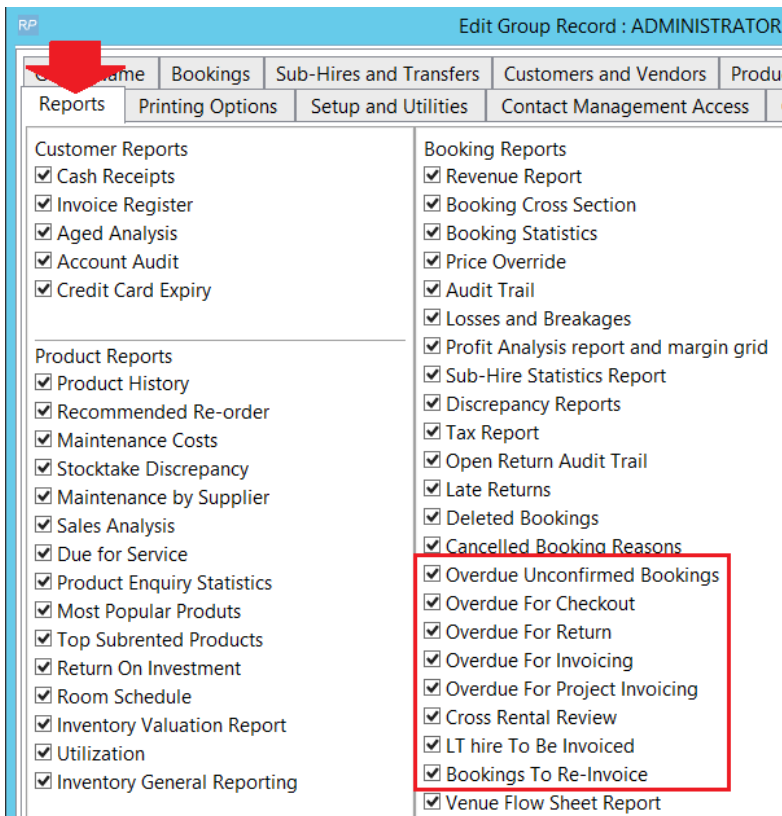
A new Cross Rental Review tab has been added to the Reminder report (v11.1.5 and higher). This new tab will display all cross rentals that contain items that are overbooked. For example, say you book qty 10 on a booking. However you only have 7, so you're now short 3. You cross rent those 3 items. Another operator receives more stock into the warehouse. So your qty avail goes from 7 to 15, which means you have enough avail to cover your original booking. In this case the cross rental is not required. The report will display the cross rental and the item(s). Warehouse staff can routinely review this report to ensure that the company saves money by not overbooking cross rented equipment.

Orphaned Cross hires and Transfers

Lists items from sub rentals and transfers, that do not have a corresponding item in a booking. This can happen if equipment on a booking is modified after a cross rental or transfer has been created for that equipment, thus rendering the transfer or cross rental unnecessary. The 'List Orphaned Cross Rentals and Transfers' option allows orphaned items to be reviewed and removed as needed, so they aren't sub rented or transferred unnecessarily.

Operator Privileges

Each report is controlled via an operator privilege. To set the operator privileges go to  and open the relevant operator group. If the privileges in the following screenshot are enabled, users from that operator group can access the report in the reminder window.

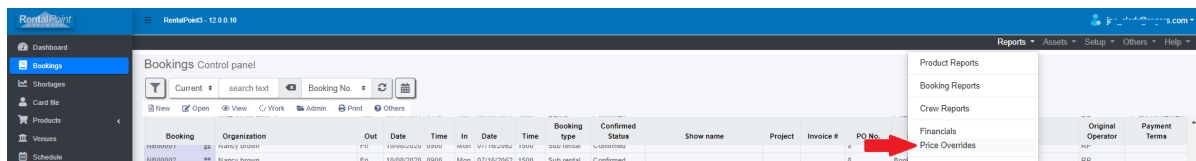



Financial Reports

Price Override

The objective of the Price Override Report is to show which operators are overriding (reducing) the calculated product price and therefore giving a discount. Every time a line item price that is calculated by the system is overridden by an operator, an entry is added to the database. The Price Override Report details all the instances that a price has been manually overridden by an operator.

To run this report, click the Price Override option from the reports menu.



 for companies that don't have set prices for many of their products, the calculated price when adding items to a booking is quite often zero, leaving the user to type in a price. This action renders the report inefficient when including zero priced items, as the report would always show that the company makes money from price overrides (since the calculated price overwritten with a price increase).

Where the objective of the report is to show which operators are overriding (reducing) the calculated price and therefore giving a discount, check the box **Exclude zero priced products** under the Options header, thus making the report more accurate.

RentalPoint
Price Overrides Report
✓ ✗

Between (Inclusive dates) and

or Select Month

Jan
 Feb
 Mar
 Apr
 May
 Jun
 Jul
 Aug
 Sep
 Oct
 Nov
 Dec

Filter

- Operator
- Customer
- Booking
- Product
- Salesperson
- All

Sequence

- Operator
- Salesperson

Options

- Include Booking Price Overrides
- Exclude zero priced products

Sample Output:

Printed: 05/14/2021
Price Override Report
Page : 1
✗

Report Dates : 05/08/2021 to 05/14/2021

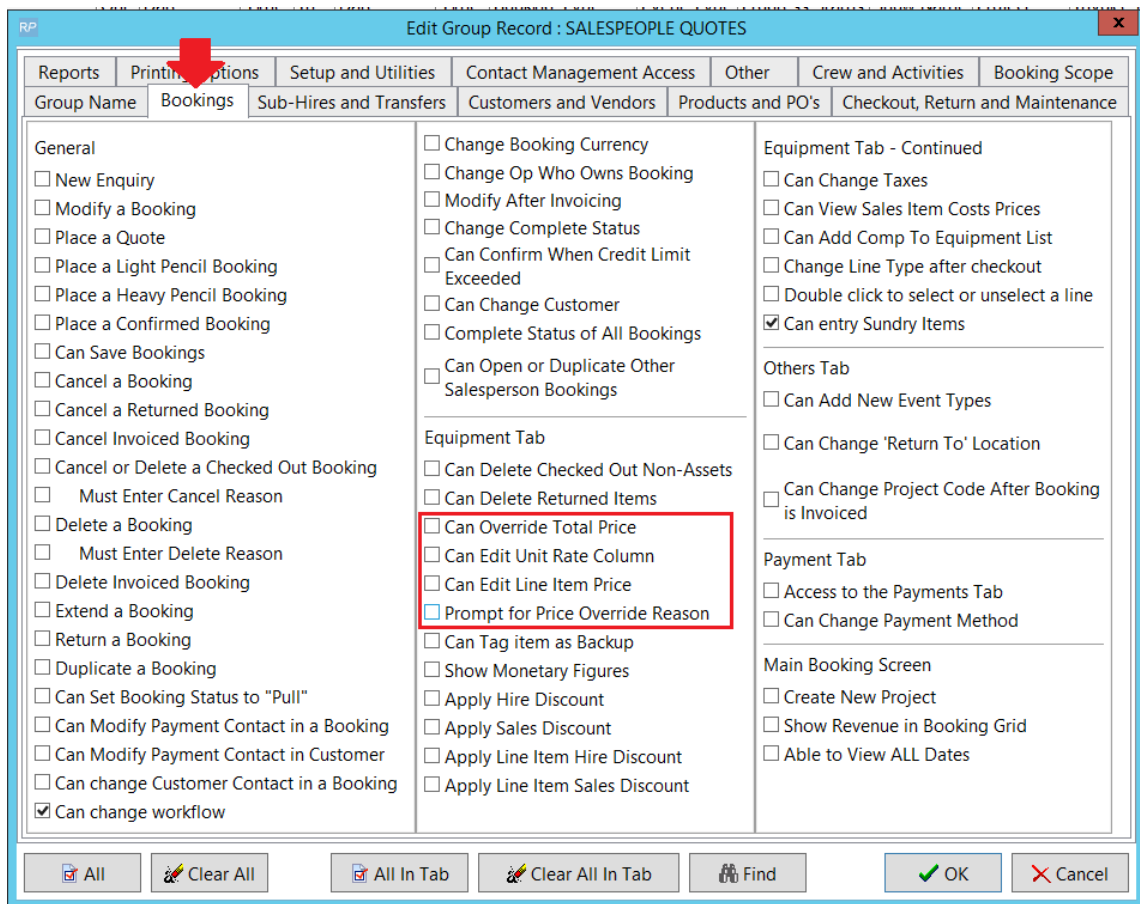
Date	Time	Operator	Booking No.	Salesperson	Product	Quantity	Price Calculated	Price Override
05/14/2021	13:11	RP	300100010	Ben William	TIN	3	405.00	350.00

Total variance of override (cost/revenue) -55.00
(price overrides have decreased revenue)

Operator Privileges

Restrict operator access to price override functionality via operator privileges. The operator privileges related to price overrides are listed below and can be found in the bookings tab of operator group permissions.:

- Can override total price
- Can edit unit rate column
- Can edit line item price
- Prompt for price override reason
-



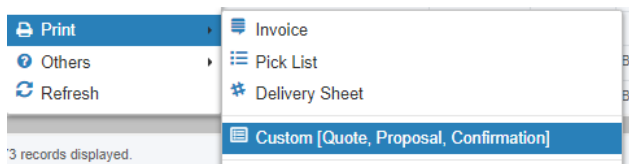
Booking Reports

Customs Document/Carnet for Booked Items

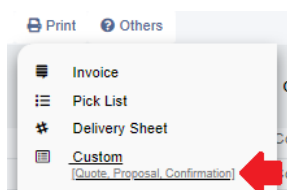
To meet international shipping requirements, RentalPoint offers functionality to build a [Virtual Carnet](#) for checked out bookings. Users can create a list of boxes/cases, allocate items to each case and either manually enter dimensions or have RentalPoint load dimensions from Inventory setup.

Sometimes a preview custom document is needed for booked items (not checked out yet). This is a simple report listing all items booked on the order and their relevant product information.

To print the report, first highlight the booking then access the right click menu



OR use the  Print icon



Cartnote

Date printed: 3/10/2022 10:13:13 AM

Job: WOWAUD00001

Event:

Qty	Description	Country of Origin	Serial No.	Value	Weight	Volume	
20	RF 15 Floor Standing Speaker	Canada		0.00	27.20	8,253.34	
1	RCF NX15 Speaker CASE			0.00	0.00	0.00	
1	RCF NX15 Speaker CASE			0.00	0.00	0.00	
1	RCF NX15 Speaker CASE			0.00	0.00	0.00	
1	RCF NX15 Speaker CASE			0.00	0.00	0.00	
1	RCF NX15 Speaker CASE			0.00	0.00	0.00	
20	NX15 10' Cable	Australia		0.00	0.00	0.00	
5	Sigma fp L Mirrorless Digital Camera			0.00	0.00	0.00	
1	Sigma fp L Mirrorless Digital Camera			0.00	0.00	0.00	
6	Sigma EVF-11 Electronic Viewfinder			0.00	0.00	0.00	
6	Sigma BP-51 Lithium-Ion Battery Pack 7.2V 1200mAh			0.00	0.00	0.00	
6	2 x Sigma Strap Holder for fp Camera			0.00	0.00	0.00	
6	Sigma HU-11 Hot Shoe Unit			0.00	0.00	0.00	
6	Sigma LCT II-TL Body Cap			0.00	0.00	0.00	
6	Sigma UAC-21 USB AC Adapter			0.00	0.00	0.00	
6	Sigma SUC-41 USB Cable			0.00	0.00	0.00	
Totals:				Items	Value	Weight	Volume
				93	0.00	544.00	165,066.81

Product Reports

Return On Investment

The **Return On Investment (ROI) report** is a condensed version of the Product History report. When detailed booking information is selected, the report will display the bookings on which the product has been used.

The **Total Cost Value** is determined by the setting configured in **Operational Parameter #64**:

- If **“asset register”** is enabled, the software uses the total of the asset purchase prices recorded for that product.
- If **“product master”** is enabled, the software uses the cost price entered on the product record multiplied by the quantity owned, ignoring any prices recorded at the individual asset level.

Nett ROI: $([\text{Income}] - [\text{Maintenance}] - [\text{Cross Rental Cost}]) / [\text{Cost Value}] \times 100$

Gross ROI: $([\text{Income}] / [\text{Cost Value}]) \times 100$

RentalPoint

Return On Investment Report

✔
✖

OK - Show report.

Between (Inclusive dates) and

or Select Month

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

Scope

Group

Category

Sub Category

Product

Select Product Leave blank for all

Output Options

Product Summary

Detailed

Group Summary

Sort By

Highest Net ROI%

Highest Gross ROI%

Lowest Net ROI%

Lowest Gross ROI%

Number of products to show

All

Show top products

Print [x](#)

Date Printed :28/05/07 Return On Investment Report Page : 1

Product : GHH15

Report Dates : 2026/05/01 to 2026/05/31 Top :100 products

Group	Category	Sub Cat.	Product Code	Description	Net ROI %	Gross ROI %	Total cost value	Amount Earned	Cross rental cost	Maint. cost
AUDIO GEAR	POWERED SPEAKERS		GHH15	GHH 15 FLOOR STANDING SPEAKER	66.67	66.67	300.00	200.00	0.00	0.00
Out Date	In Date	Trans Type	Booking No	Qty Rented	Total Price	Days	Daily Price	Booking Discount	Amount After Discount	
2026/04/30	2026/05/01	Hire	SPEN00027	1	100.00	1	100.00	0.00%	100.00	
2026/04/30	2026/05/01	Hire	SPEN00027	1	100.00	1	100.00	0.00%	100.00	

Report Legend

Net ROI Calculation is ((Income)-[Maintenance]-[Cross rental cost]/[Cost value]) x 100.
Gross ROI Calculation is (Income/[Cost value]) x 100.

Utilization Report

Available from the Dashboard Products Card

The screenshot shows the RentalPoint3 dashboard interface. On the left is a navigation menu with items: Dashboard, Bookings, Shortages, Card file, Products, Venues, Schedule, Action Lists, and Contacts. The main content area is titled 'Products card' and includes a date range selector (4/27/2019 to 4/27/2022), a refresh button, and a dropdown menu set to '0 - Main'. Below this is a 'Products Utilization Chart - Top 10' which is a donut chart. A legend to the left of the chart lists ten product codes with corresponding color swatches: MXPRO (red), TS315CAS (green), ALTS315 (orange), RCFNX15 (cyan), TVBRAK (blue), TS315CAB (teal), SAMIND (pink), SAMTVREM (dark blue), RF123 (orange-red), and NX1510 (brown). At the bottom of the dashboard, there is a teal bar showing 'Total value of assets' as 49,277.4 with a shopping cart icon. A 'More' button is located at the bottom right of the dashboard area.

OR from the Reports menu

RentalPoint3 DB [12.0.6.0] Rev [3262]

Reports

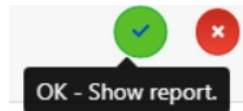
Index to Reports
Action Lists
Report Builder

Booking Reports Crew Reports Financial Reports Product Reports Sub hire Reports

Utilisation Report

Product Listing Report

Return On Investment Report



Choose selection criteria for your report, then click

Utilisation Report

Between (Inclusive dates) 28/04/2021 and 27/04/2022
or Select Month
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

Scope

Group
 Category
 Sub Category
 Product
Select Code Leave blank for all

Sort By

Highest % Utilisation # of products (Most Used)
 Lowest % of Utilisation # of products (Least Used)
 All

Location

All
 Select Location 0 - Main

Confirmation status

Quote
 Light Pencil
 Heavy Pencil
 Confirmed

Location : 0 - Main
Report Dates : 28/04/2021 to 27/04/2022

Group	Category	Sub Cat.	Code	Description	# times Rented	Qty Owned	Cost Price	Util %	Max Days	Days rented
AUDIO	POWSPEAK		MXPRO	MX Processor	44	1	60.00	49.86	365	44
AUDIO	SPKCASE		TS315CAS	Alto TS315 Speaker CASE	24	6	0.00	7.63	2190	24
AUDIO	POWSPEAK		ALTS315	Alto TS315 Speaker	24	8	37.50	6.13	2920	24
AUDIO	POWSPEAK		RCFNX15	RF 15 Floor Standing Speaker	12	3	0.00	5.57	1095	12
VIDEO	TV		TVBRAK	TV Adjustable Wall Bracket	33	15	0.00	4.49	5475	33
AUDIO	SPKCABLE		TS315CAB	TS315 10' Cable	24	12	50.00	3.88	4380	24
VIDEO	TV		SAMIND	60" Samsung LED Touchscreen	12	7	1,714.29	3.56	2555	132
VIDEO	TV		SAMTVREM	Samsung TV Remote	33	20	0.00	3.45	7300	33
AUDIO	POWSPEAK		RF123	RF 15 Floor Standing Speaker	9	2	1,000.00	3.15	730	9
G1	SPKCABLE		NX1510	NX 15 10' Cable	17	10	0.00	1.89	3650	17
AUDIO	SPKCASE		NX15CASE	RCF NX15 Speaker CASE	21	12	26.67	1.67	4380	21
G2	C53		SAM60	60" Samsung LED Touchscreen	20	17	470.59	0.98	6205	20
COMPUTER	NOTEBOOK		5160	Dell 5160 Notebook Computer	1	5	4,850.00	0.82	1825	15
VIDEO	CAMERA		SIGFP	Sigma fp L Mirrorless Digital Camera	3	4	0.00	0.75	1460	3
VIDEO	CAMACC		BP51	Sigma BP-51 Lithium-Ion Battery Pack 7.2V 1200mAh	3	5	0.00	0.60	1825	3

Report Title is driven by the terminology parameter for English or US spelling (fig 1.1 below)

- "Utilization" non-English spelling
- "Utilisation" - English spelling

Dates on the view are *inclusive* in respect to the end date (whereas the end date in RP2 is exclusive)

- Inclusive – transactions for the end date will be included
- Exclusive – transactions for the end date are not included

of Times Rented = Number of days the product was used within the specified time period.

Quantity Owned = Number of items owned by your company.

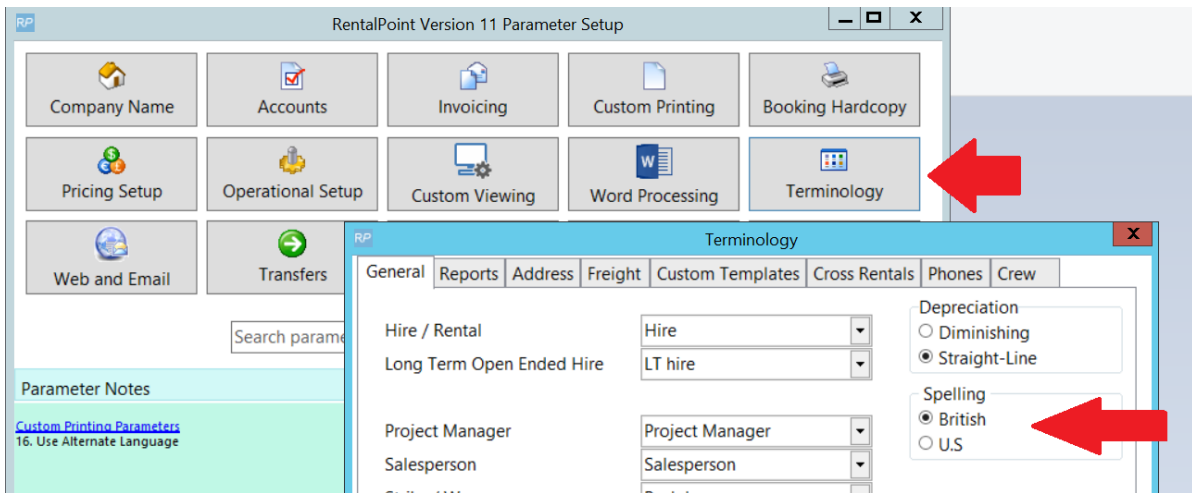
Cost Price = Amount the item was purchased for.

Util % = the amount of time the item was rented in the specified time period, expressed as a percentage.

Max Days = the maximum number of days the item is available within the specified time period. (Days x Quantity Owned)

Days rented = the number of days between the warehouse out and in dates for the product for the booking

Fig 1.1



Fast Report Templates

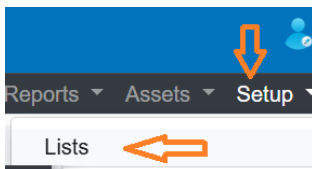
About

Merge your data with pre-designed templates to provide quotes, invoices picklists etc.

To make changes to templates, first copy the template to a new name, then make changes to the copy.

Contact support@rentp.com for assistance with custom template modifications.


Access via:




RentalPoint3 DB [12.0.23.17] Rev [6641]

- Dashboard
- Bookings
- Shortages
- Card file
- Products
- Venues
- Schedule
- Crew Planner
- Purchase Orders
- Action Lists
- Contacts

Lists





















- Activity Types
- Cancellation Reasons
- Crew Portal Access
- Currencies
- Divisions
- Email Signatures
- Email Paragraphs
- Expense Codes
- Fast Reports 

Copy

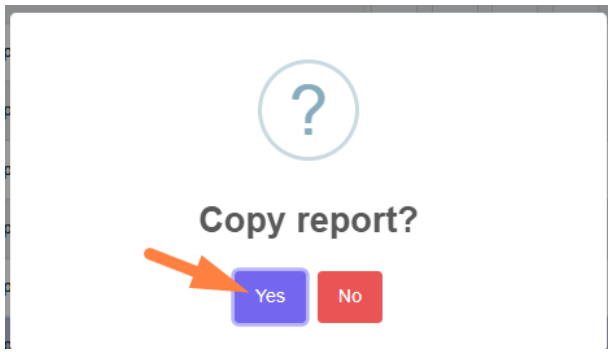
Use the  icon to copy any template to a new name

RentalPoint3 DB [12.0.23.17] Rev [6641]

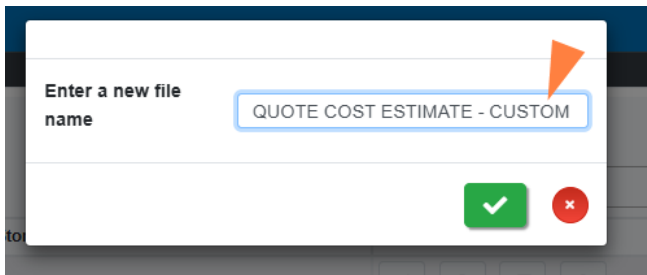
Show 10 entries Search:

Report Name	Report Type	Stored Procedure	Actions
Project Invoice - Default	Project Invoice	Frp_Report_ProjInvoice	   
Project Quote	Project Quote	Frp_Report_ProjQuote	   
Quote - Default	Quote	Frp_Report_Quote	   
Quote Alternate	Quote	Frp_Report_Invoice_Alt	   
Quote Cost Estimate	Quote	Frp_Report_QuotesCostEstimate	   

Showing 51 to 60 of 98 entries Previous 1 ... 5 **6** 7 ... 10 Next



Change the name as needed



The new template will appear alphabetically sorted in the Fast Report Template window

RentalPoint3 DB [12.0.23.17] Rev [6641]

























Show 10 entries Search:

Report Name	Report Type	Stored Procedure	Actions
Project Invoice - Default	Project Invoice	Frp_Report_ProjInvoice	
Project Quote	Project Quote	Frp_Report_ProjQuote	
Quote - Default	Quote	Frp_Report_Quote	
Quote Alternate	Quote	Frp_Report_Invoice_Alt	
Quote Cost Estimate	Quote	Frp_Report_QuotesCostEstimate	
Quote Cost Estimate - CUSTOM	Quote	Frp_Report_QuotesCostEstimate	

Showing 51 to 60 of 99 entries Previous 1 ... 5 6 7 ... 10 Next

Edit

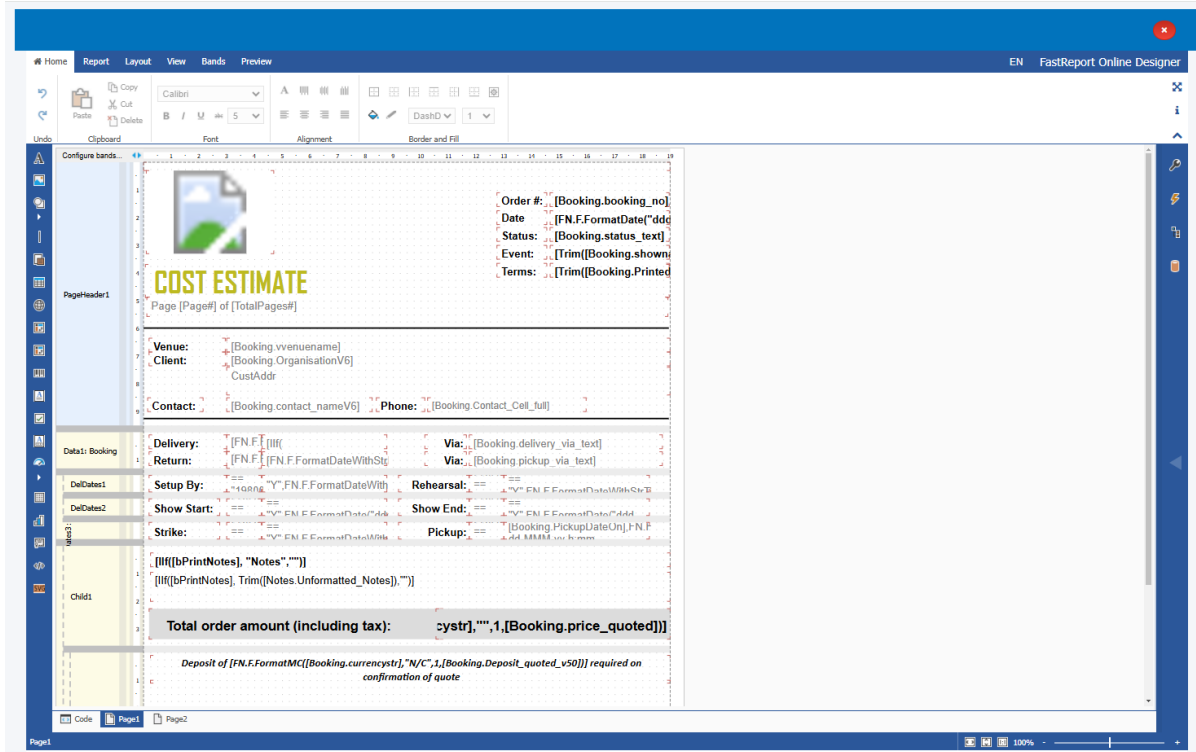
Use the icon to edit your custom template

Report Name	Report Type	Stored Procedure	Actions
Project Invoice - Default	Project Invoice	Frp_Report_ProjInvoice	   
Project Quote	Project Quote	Frp_Report_ProjQuote	   
Quote - Default	Quote	Frp_Report_Quote	   
Quote Alternate	Quote	Frp_Report_Invoice_Alt	   
Quote Cost Estimate	Quote	Frp_Report_QuotesCostEstimate	   
Quote Cost Estimate - CUSTOM	Quote	Frp_Report_QuotesCostEstimate	   

Showing 51 to 60 of 99 entries

Previous 1 ... 5 6 7 ... 10 Next

This opens the *Fast Report Online Designer* where the layout of the report can be manipulated. [<More information>](#)

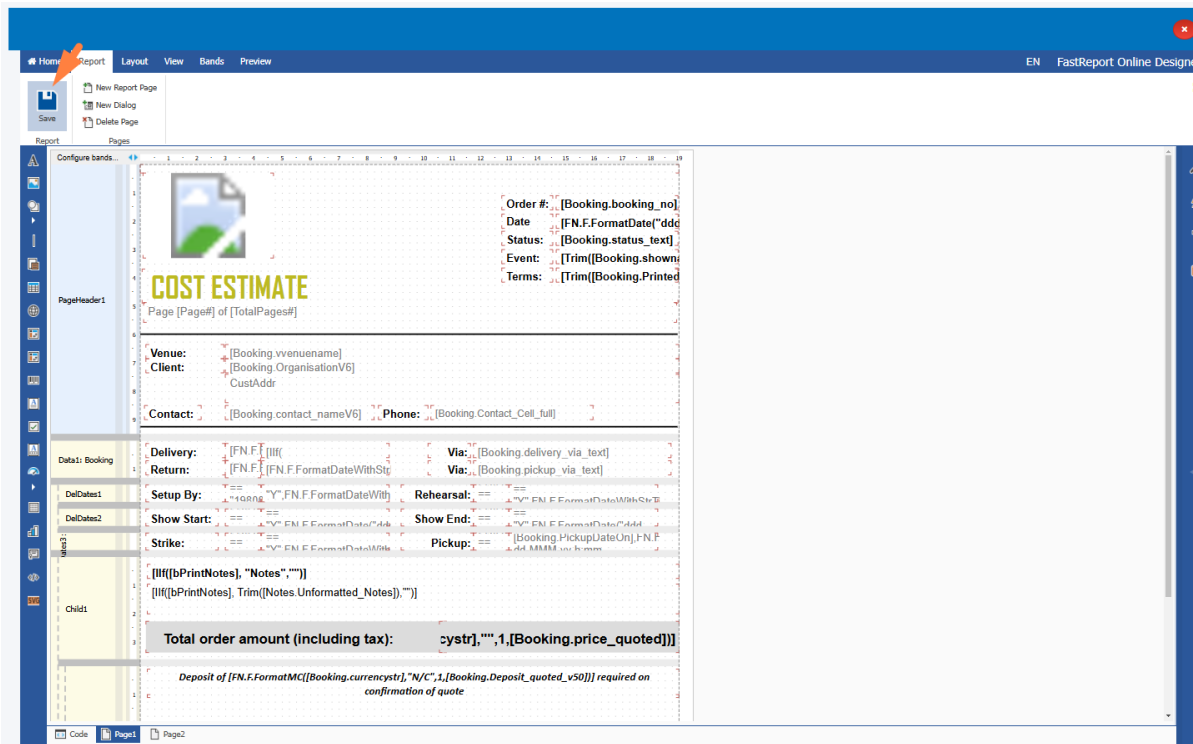


The screenshot shows the 'FastReport Online Designer' interface. The main workspace displays a report layout for 'COST ESTIMATE'. The report includes a header section with the title 'COST ESTIMATE' and a page number 'Page [Page#] of [TotalPages#]'. Below the header, there are several data fields and sections:

- Venue:** [Booking.vvenueName]
- Client:** [Booking.OrganisationV6], CustAddr
- Contact:** [Booking.contact_nameV6], **Phone:** [Booking.Contact_Cell_full]
- Delivery:** [FN.F.FormatDate]([Booking.delivery_date]) **Via:** [Booking.delivery_via_text]
- Return:** [FN.F.FormatDateWithStg]([Booking.return_date]) **Via:** [Booking.pickup_via_text]
- Setup By:** [FN.F.FormatDateWithStg]([Booking.Setup_By]) **Rehearsal:** [FN.F.FormatDateWithStg]([Booking.Rehearsal])
- Show Start:** [FN.F.FormatDate]([Booking.Show_Start]) **Show End:** [FN.F.FormatDate]([Booking.Show_End])
- Strike:** [FN.F.FormatDateWithStg]([Booking.Strike]) **Pickup:** [Booking.PickupDateUn]([Booking.PickupDateUn])

The report also includes a section for notes: `[[[([bPrintNotes, "Notes", ""])]]]` and a total order amount: `Total order amount (including tax): [cystri], "", 1, [Booking.price_quoted]]`. A deposit note is also present: `Deposit of [FN.F.FormatMC([Booking.currencycystri], "N/C", 1, [Booking.Deposit_quoted_v50])] required on confirmation of quote`. The interface includes a menu bar (Home, Report, Layout, View, Bands, Preview), a toolbar with various editing tools, and a right-hand sidebar with additional options.

Remember to save your changes when finished




A note on adding data not already available in the data list

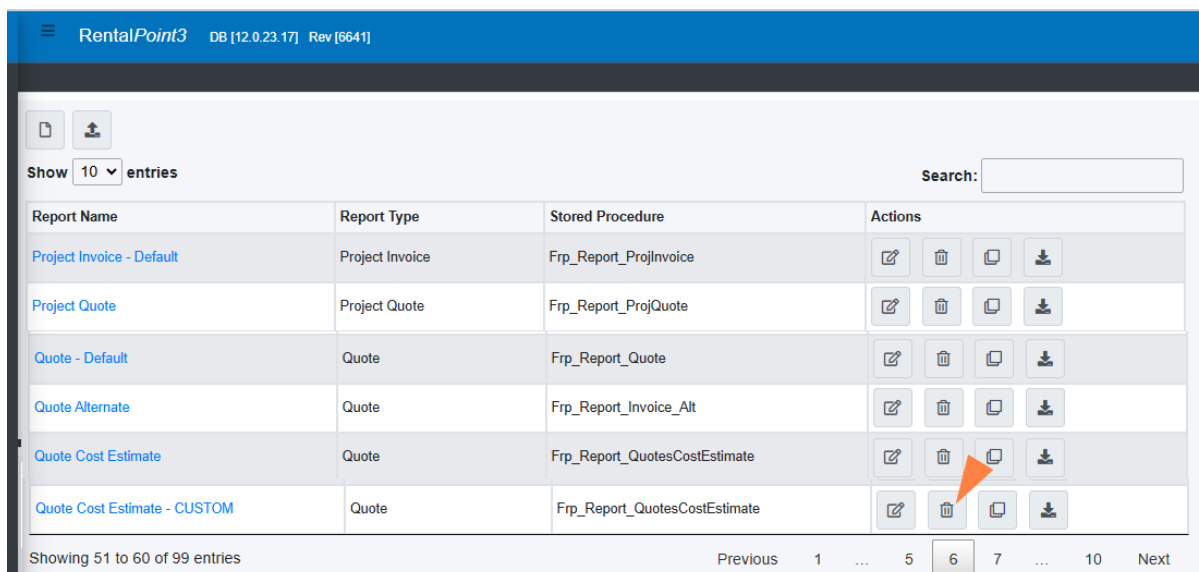
Additional fields must be made available via the stored procedure associated with the database.

Please contact support@rentp.com for assistance.

Delete

Use the  option to delete unwanted custom reports.

Deleted reports can only be restored from backup, so consider deactivating your template instead

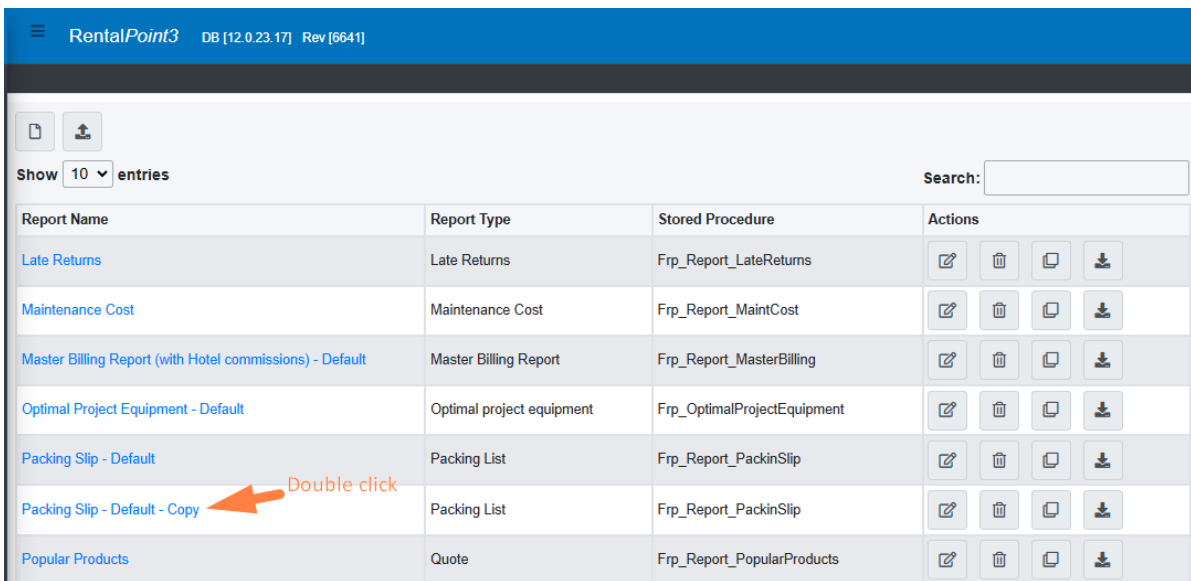


Hide

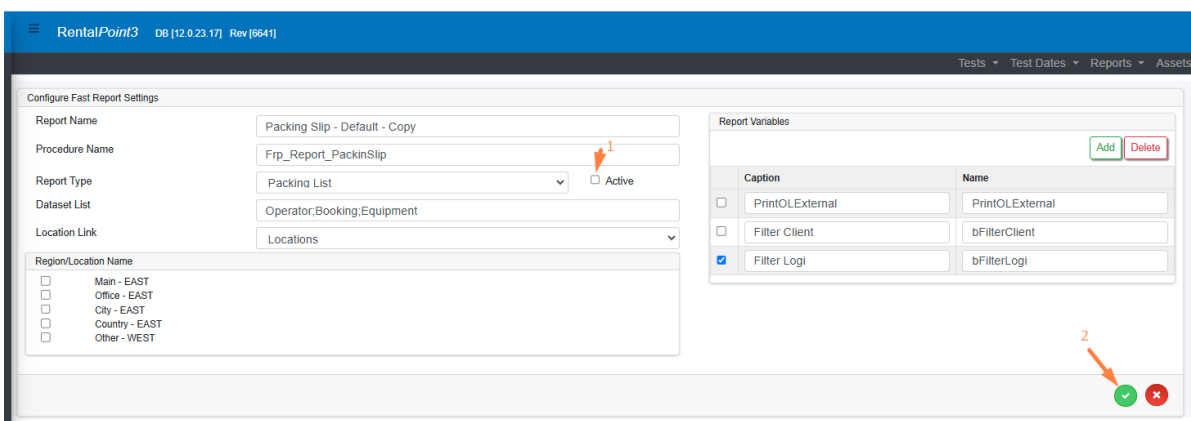
To hide your template from any selection list, uncheck Active in report settings



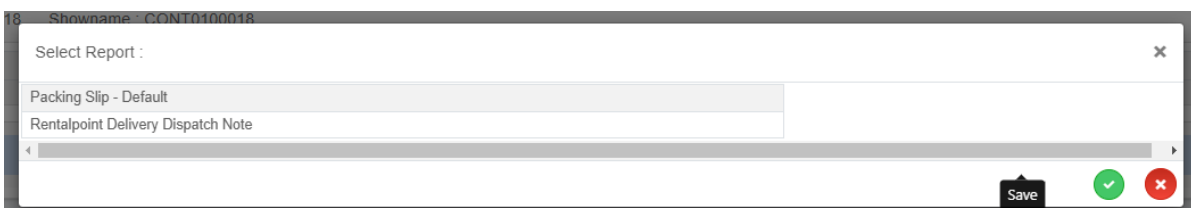
Goto Setup-->Lists-->Fast Reports, then double click the template name



uncheck 'Active' and save



The template will no longer show up as an option



SQL Queries

The Queries provided below are written in SQL to pull information from your RentalPoint database. The queries can be run in SQL Server Management Studio OR by using the *Report Builder* in RentalPoint3

Assets Checked Out to a Booking

Show assets checked out to the booking MYTEST00016

Change the value of @Booking_no below and execute this query

```
DECLARE @BOOKING_NO VARCHAR(13)
SET @BOOKING_NO = 'MYTEST00016'
```

```
SELECT O.Booking_no as [Booking Number],
       B.showName as [SHOWNAME],
       B.ddate as [WH OUT],
       B.rdate as [WH IN],
```

```
       O.Product_code as [Product Code],
       A.DESCRPTION as [Asset Description],
       A.ASSET_CODE as [Barcode],
       A.SERIAL_NO AS [Serial #],
       CAST(A.Cost as decimal(9,2)) as [Cost]
```

```
FROM tblAssetran O INNER JOIN tblasset01 A ON A.product_code = O.Product_code and A.stock_number = O.stock_number
INNER JOIN tblbookings B on O.Booking_no = B.booking_no
LEFT OUTER JOIN TBLINVMAS M ON M.product_code = O.product_code
Where O.Booking_no = @BOOKING_NO
order by M.SEQ_NO
```

Assets NOT Scanned in Stocktake Since Date Specified

This query can be run to find all the assets that have not be scanned in 12 months or 24 months in a stocktake, so we know the missing items. You'll need to change the value of EntryDateTime bolded below to get the records within your specified timeframe.

```
SELECT A1.*
FROM tblasset01 A1
WHERE IsNull(A1.asset_code, "") <> "
       AND A1.asset_code NOT IN (SELECT DISTINCT A.asset_code
                                FROM tblasset01 A
                                INNER JOIN tblstocktakhistory S
                                    ON S.product_code =
                                       A.product_code
                                   AND S.stock_number =
                                       A.stock_number)
```

WHERE entrydatetime < '2025-12-31')

Assets NOT Scanned Out Since Date

In search of unused assets?

This query will show assets that have not be scanned on to or off a booking, thus haven't been used in RentalPoint.

```
SELECT A1.*
FROM tblasset01 A1
WHERE Isnull(A1.asset_code, "") <> ""
AND A1.asset_code NOT IN (SELECT DISTINCT A.asset_code
FROM tblasset01 A
INNER JOIN tblassetran S
ON S.product_code =
A.product_code
AND S.stock_number =
A.stock_number)
```

Assets Out On A Given Date

The query you have will give you assets out on a given date - even if they were checked back in after that date Technically they were OUT on the given date, it will also show items that were never returned.

Things to Note:

- ActOutDate is the date the item was checked out
- ActInDate is the date the item was checked in
- The 1980 showing in ActInDate would mean the item was never checked back in. This could be a long term hire/an asset that was never returned or returned but not checked back in.
- A.DisDate is the disposal date of the asset, the queries below will exclude assets disposed of

Change the value of @SingleDate (bolded below) to suit your needs

```
DECLARE @SingleDate AS DATETIME
```

```
SET @SingleDate = '20210101'
```

```
SELECT A.product_code          [Product Code],
A.asset_code                  [Barcode],
A.description                  [Asset Description],
AT.booking_no                 [Out on booking],
B.showname                    [Showname],
B.organizationv6              [Customer],
CONVERT(VARCHAR, b.ddate, 101) [Booking Out Date],
CONVERT(VARCHAR, b.rdate, 101) [Booking In Date],
CONVERT(VARCHAR, AT.actoutdate, 101) [Asset Out Date],
CASE
WHEN AT.actindate = Cast('Jan 1 1980' AS DATETIME) THEN 'Still Out'
WHEN AT.actindate > Cast('Jan 1 1980' AS DATETIME) THEN
CONVERT(VARCHAR, AT.actindate, 101)
END
AS [Asset Return Date]
FROM tblassetran AT
LEFT OUTER JOIN vwbookandhist B
```

```

        ON B.booking_no = AT.booking_no
LEFT OUTER JOIN tblcust C
        ON B.custcode = C.customer_code
INNER JOIN tblasset01 A
        ON A.product_code = AT.product_code
        AND A.stock_number = AT.stock_number
WHERE ( ( AT.actoutdate > Cast('Jan 1 1980' AS DATETIME) )
        AND At.actoutdate <= @SingleDate )
AND ( ( AT.actindate = Cast('Jan 1 1980' AS DATETIME) )
        OR ( AT.actindate > @SingleDate ) )
AND ( Isnull(a.disdate, '1980-01-01') = '1980-01-01' )
/*assets are not sold*/
ORDER BY A.product_code,
        A.asset_code,
        AT.booking_no

```

Assets Out Today

The query you have will give you assets out on a given date - even if they were checked back in after that date Technically they were OUT on the given date, it will also show items that were never returned.

Things to Note:

- ActOutDate is the date the item was checked out
- ActInDate is the date the item was checked in
- The 1980 showing in ActInDate would mean the item was never checked back in. This could be a long term hire/an asset that was never returned or returned but not checked back in.
- A.DisDate is the disposal date of the asset, the queries below will exclude assets disposed of

```

DECLARE @SingleDate AS DATETIME

```

```

SET @SingleDate = GETDATE()

```

```

SELECT  A.product_code [Product Code],

```

```

        A.asset_code [Barcode],

```

```

        A.DESCRPTION [Asset Description],

```

```

        AT.booking_no [Out on booking],

```

```

        AT.ActOutDate [Out Date],

```

```

        AT.ActInDate [In Date],

```

```

        A.COST [Cost]

```

```

FROM tblassetran AT INNER JOIN tblasset01 A ON A.product_code = AT.product_code and A.stock_number = AT.stock_number

```

```

WHERE ((AT.ActOutDate > CAST('Jan 1 1980' AS DateTime)) and At.ActOutDate <= @SingleDate)

```

```

        AND ((AT.ActInDate = CAST('Jan 1 1980' AS DateTime)) OR (AT.ActInDate > @SingleDate))

```

```

AND ( ISNULL(a.DisDate, '1980-01-01') = '1980-01-01' ) /*assets are not sold*/

```

```

ORDER BY A.product_code, A.asset_code, AT.booking_no

```

Asset Out Between Date Range

The query you have will give you assets out between the selected date range

Things to Note:

- ActOutDate is the date the item was checked out
- ActInDate is the date the item was checked in
- The 1980 showing in ActInDate would mean the item was never checked back in. This could be a long term hire/an asset that was never returned or returned but not checked back in.
- A.DisDate is the disposal date of the asset, the queries below will exclude assets disposed of

Change the value of @FromDate and @ToDate (bolded below) to whatever you want it to be and then copy everything from below this line into your query window.

```
DECLARE @FromDate AS DATETIME
DECLARE @ToDate AS DATETIME
SET @FromDate = '20201201'
SET @ToDate = '20210131'
SELECT A.product_code [Product Code],
       A.asset_code [Barcode],
       A.description [Asset Description],
       AT.booking_no [Out on booking],
       B.showname [Showname],
       B.organizationv6 [Customer],
       convert(varchar,b.ddate,101) [Booking Out Date],
       convert(varchar,b.rdate,101) [Booking In Date],
       convert(varchar,AT.ActOutDate,101) [Asset Out Date],
       CASE
       WHEN AT.actindate = CAST('Jan 1 1980' AS datetime) THEN 'Still Out'
       WHEN AT.ActInDate > CAST('Jan 1 1980' AS DateTime) THEN convert(varchar,AT.ActInDate,101)
       END AS [Asset Return Date]
FROM tblasstran AT
LEFT OUTER JOIN vwbookandhist B
       ON B.booking_no = AT.booking_no
LEFT OUTER JOIN tblcust C
       ON B.custcode = C.customer_code
INNER JOIN tblasst01 A
       ON A.product_code = AT.product_code
       AND A.stock_number = AT.stock_number
WHERE ( ( AT.actoutdate BETWEEN @FromDate AND @ToDate )
```

OR (AT.actindate BETWEEN @FromDate AND @ToDate)

ORDER BY At.actoutdate,

at.actindate

Assets Out to Maintenance

Shows a list of all assets checked out to maintenance

```
SELECT IM.groupfld [Group],
       IM.category [Category],
       A.product_code [Product],
       A.asset_code [Barcode],
       A.serial_no [Serial],
       A.locn [Item Location],
       A.cost,
       A.est_resale,
       A.insured_val,
       CASE
         WHEN A.servicestatus = 0 THEN 'Active'
         WHEN A.servicestatus = 1 THEN 'Temporarily Out of Service'
         ELSE 'Permanently Out of Service'
       END [Status]
FROM tblasasset01 A
     LEFT OUTER JOIN tblinvmas IM
       ON IM.product_code = A.product_code
WHERE A.servicestatus = 1
ORDER BY IM.groupfld,
         IM.category,
         A.product_code,
         A.asset_code
```

Assets Not Returned

Lists all asset Tracked items checked out to a booking where the return date of the booking has passed and the items are still out

```
SELECT I.category [Category],
       OA.product_code [Product Code],
       OA.qtycheckedout [Qty Out],
       OA.qtyreturned [Qty Ret],
       A.asset_code [Asset Barcode],
       A.description [Asset Description],
       OA.booking_no [Booking Number],
       B.showname [Showname],
       b.ddate [WH Out Date],
       b.rdate [WH In Date]
FROM tblassetran OA
     LEFT OUTER JOIN tblbookings B
       ON B.booking_no = OA.booking_no
     LEFT OUTER JOIN tblinvmas I
       ON I.product_code = OA.product_code
     LEFT OUTER JOIN tblasasset01 A
       ON A.product_code = OA.product_code
       AND A.stock_number = OA.stock_number
     LEFT OUTER JOIN tblitemtran T
```

```

        ON T.booking_no_v32 = OA.booking_no
        AND T.subrentallinkid = OA.itemtranid
WHERE B.rdate < Getdate()
    AND OA.stock_number <> 65535
    AND ( OA.qtycheckedout > OA.qtyreturned )
    AND T.trans_type_v41 <> 6
ORDER BY I.category,
        B.rdate

```

Asset Movement Including Last Stock Take

```

SELECT A.asset_code           [Barcode],
       A.product_code        [Product],
       A.description,
       A.location,
       A.stock_number        [Stock Number],
       A.serial_no           [Serial],
CASE
    WHEN idisposalttype = 0 THEN 'CURRENT'
    WHEN idisposalttype = 1 THEN 'SOLD'
    WHEN idisposalttype = 2 THEN 'LOST'
    WHEN idisposalttype = 3 THEN 'STOLEN'
    WHEN idisposalttype = 4 THEN 'WRITTENOFF'
END,
       A.cost,
       A.est_resale,
       A.insured_val,
(SELECT TOP 1 CASE
WHEN AT.actindate = Cast('Jan 1 1980' AS DATETIME) THEN
    'Out'
WHEN AT.actindate > Cast('Jan 1 1980' AS DATETIME) THEN
    'In'
END
FROM tblasstran AT
WHERE AT.product_code = A.product_code
    AND AT.stock_number = A.stock_number
ORDER BY actoutdate DESC) [Last Asset Movement Type],
(SELECT TOP 1 AT.booking_no
FROM tblasstran AT
WHERE AT.product_code = A.product_code
    AND AT.stock_number = A.stock_number
ORDER BY actoutdate DESC)
[Last Asset Movement Booking],
(SELECT TOP 1 CASE
WHEN AT.actindate = Cast('Jan 1 1980' AS DATETIME) THEN
    'Still Out On Booking'
ELSE CONVERT(VARCHAR, AT.actindate, 103)
END
FROM tblasstran AT
WHERE AT.product_code = A.product_code
    AND AT.stock_number = A.stock_number
ORDER BY actoutdate DESC) [Last Asset Movement Date],
(SELECT Max(entrydatetime)
FROM tblstocktakhistory S
WHERE A.product_code = S.product_code
    AND A.stock_number = S.stock_number) [Last StockTake],
CASE
WHEN Len(b.rdate) > 0 THEN CONVERT(VARCHAR, B.rdate, 103)
ELSE ''
END [Warehouse IN Date]

```

```

FROM tblasset01 A
  LEFT OUTER JOIN tblinvmas IM
    ON IM.product_code = A.product_code
  LEFT OUTER JOIN tblbookings B
    ON B.booking_no = A.booking_no
/*WHERE A.Asset_Code = 'XXXXXX' */
ORDER BY IM.groupFld,
  IM.category,
  A.product_code,
  A.asset_code

```

Asset Movements Including Number of Times Rented

```

SELECT IM.groupFld [Group],
IM.category [Category],
A.PRODUCT_CODE [Product],
A.ASSET_CODE [Barcode],
A.SERIAL_NO [Serial],
A.locn [Item Location],
A.COST,
A.EST_RESALE,
A.INSURED_VAL,
A.ServiceStatus,
A.iDisposalType,
CASE
WHEN (A.ServiceStatus = 0
AND ISNULL(A.iDisposalType,0) = 0) THEN 'Active'
WHEN A.ServiceStatus = 1 THEN 'Temporarily Out of Service'
ELSE 'Permanently Out of Service'
END [Status],
CASE
WHEN A.iDisposalType = 1 THEN 'SOLD'
WHEN A.iDisposalType = 2 THEN 'Lost/Missing'
WHEN A.iDisposalType = 3 THEN 'Stolen'
WHEN A.iDisposalType = 4 THEN 'Written Off'
ELSE ''
END [Disposal Type],

(SELECT TOP 1 CASE
WHEN AT.ActInDate = CAST('Jan 1 1980' AS DateTime) THEN 'Out'
WHEN AT.ActInDate > CAST('Jan 1 1980' AS DateTime) THEN 'In'
END
FROM tblAssetran AT
WHERE AT.product_code = A.PRODUCT_CODE
AND AT.stock_number = A.STOCK_NUMBER
ORDER BY ActOutDate DESC) [Last Asset Movement Type],

(SELECT TOP 1 AT.booking_no
FROM tblAssetran AT
WHERE AT.product_code = A.PRODUCT_CODE
AND AT.stock_number = A.STOCK_NUMBER
ORDER BY ActOutDate DESC) [Last Asset Movement Booking],

((SELECT B.showname from tblbookings B where B.booking_no = (SELECT TOP 1 AT.booking_no
FROM tblAssetran AT
WHERE AT.product_code = A.PRODUCT_CODE
AND AT.stock_number = A.STOCK_NUMBER
ORDER BY ActOutDate DESC))
UNION

```

```
(SELECT B.showname from tblHistbks B where B.booking_no = (SELECT TOP 1 AT.booking_no
FROM tblAssetran AT
WHERE AT.product_code = A.PRODUCT_CODE
AND AT.stock_number = A.STOCK_NUMBER
ORDER BY ActOutDate DESC))) [Showname],
```

```
(SELECT TOP 1 CASE
WHEN AT.ActInDate = CAST('Jan 1 1980' AS DateTime) THEN 'Still Out On Booking'
ELSE Convert(varchar, AT.ActInDate,101)
END
FROM tblAssetran AT
WHERE AT.product_code = A.PRODUCT_CODE
AND AT.stock_number = A.STOCK_NUMBER
ORDER BY ActOutDate DESC) [Last Asset Movement Date],
```

```
(SELECT COUNT(AT.ID)
FROM tblAssetran AT
LEFT OUTER JOIN tblbookings B ON B.booking_no = AT.booking_no
WHERE AT.product_code = A.PRODUCT_CODE
AND AT.stock_number = A.STOCK_NUMBER /* AND B.booking_type_v32 in (0, 4, 6, 14, 15, 16, 17)*/) [Number Times Rented],
```

```
(SELECT ISNULL(SUM(ABS(DATEDIFF(DAY, AT.ActOutDate, AT.ActInDate))), 0)
FROM tblAssetran AT
LEFT OUTER JOIN tblbookings B ON B.booking_no = AT.booking_no
WHERE AT.product_code = A.PRODUCT_CODE
AND AT.stock_number = A.STOCK_NUMBER
AND AT.ActInDate > AT.ActOutDate /* must be returned in order to calculate correctly */ /* AND B.booking_type_v32 in (0, 4, 6, 14,
15, 16, 17) */) [Number Days Rented]
FROM tblAsset01 A
LEFT OUTER JOIN tblInvmas IM ON IM.product_code = A.PRODUCT_CODE
/*WHERE a.iDisposalType IN (2) */

/*WHERE A.Asset_Code = 'XXXXXX' */
ORDER BY IM.groupFld,
IM.category,
A.PRODUCT_CODE,
A.ASSET_CODE
```

Assets Purchased Between Date Range

Provides a list of assets between the bolded dates below. Change the value of the dates to suit your criteria.

```
SELECT IM.groupfld AS [Product Group],
IM.category AS [Product Category],
A.product_code AS [Product Code],
IM.descriptionv6 AS [Product Description],
A.asset_code AS [Bar Code Number],
A.description AS [ASSET Description],
A.serial_no AS [Serial #],
A.cost AS [Cost Price],
```

```

w.Locn_name AS [Warehouse Location],
H.locn_name AS [Home Location],
A.purdate AS [Purchase Date],
CAST(disposal_amt as decimal(9,2)) AS [Disposal Amount],
disdate AS [Disposal Date]
FROM tblasset01 AS A
LEFT OUTER JOIN tblinvmas AS IM
ON IM.product_code = A.product_code
LEFT OUTER JOIN tblLocnlist W on W.Locn_number = a.locn
LEFT OUTER JOIN tblLocnlist H on H.Locn_number = a.HomeLocn
WHERE A.purdate BETWEEN '2008-01-01' AND '2022-12-31'
ORDER BY IM.groupfld,
IM.category,
A.product_code,
A.asset_code

```

Assets Entered Since Date

Shows a list of all assets entered since a specific purchase date. Change the purchase date below (bolded below) for specific results.

```

select A.ASSET_CODE as [Barcode],
A.DESCRPTION as [Asset Description],
A.modelNumber as [Asset Model #],
A.PurDate as [Purchase Date],
A.PRODUCT_CODE as [Product Code],
I.descriptionV6 as [Product Description]
from tblasset01 A
Left Outer Join tblinvmas I on I.product_code = A.PRODUCT_CODE
where A.purdate >= '2016-04-01'

```

Asset Listing

Lists Assets in inventory with some rate and product information

```

SELECT A.description                AS [Asset Description],
       A.asset_code                 AS [Barcode],
       Cast(R.rate_1st_day AS DECIMAL(12, 2)) AS [Rate1st Day],
       Cast(R.rate_week AS DECIMAL(12, 2)) AS [Rate Week],
       Cast(R.rate_long_term AS DECIMAL(12, 2)) AS [Long Term Rate],
       Cast(R.replacementvalue AS DECIMAL(12, 2)) AS [Replacement Value],
       IM.product_code              AS [Product_code],
       A.serial_no                  AS [Serial #],
       A.modelnumber                AS [Model No.]
FROM   tblasset01 A
LEFT OUTER JOIN tblinvmas IM
        ON A.product_code = IM.product_code
LEFT OUTER JOIN dbo.tblratetbl AS R
        ON R.productcode = IM.product_code
        AND R.tableno = 0
ORDER BY IM.product_code,
         A.asset_code

```

Assets Overdue for Return

```

Select B.booking_no 'Booking #',

       CASE WHEN B.Booking_type_v32 = 0 THEN 'Rental'

           WHEN B.booking_type_v32 = 1 THEN 'Sub Hire'

           WHEN B.booking_type_v32 = 2 THEN 'Quote'

       ELSE " END AS Type,

       B.dDate 'Warehouse Out',

       B.rDate 'Warehouse In',

       t.product_code 'Product Code',

       a.DESCRPTION 'Description',

       t.stock_number 'Stock #',

       a.ASSET_CODE 'Barcode',

       a.RFIDTag 'RFID',

       l.Locn_name 'Location',

       a.LOCATION 'Bin Location'

from tblAssetran t

left outer join tblItemtran l on l.SubRentalLinkID = t.ItemTranID

left outer join tblbookings b on b.booking_no = l.booking_no_v32

left outer join tblasset01 A on a.PRODUCT_CODE = t.product_code and a.STOCK_NUMBER = t.stock_number

left outer join tblLocnlist l on l.Locn_number = a.locn

where b.rdate <= getdate() and t.stock_number <> '65535' and t.ActInDate = '1980-01-01 00:00:00.000' and booking_type_v32 in
(0,1,2,5)

order by b.rDate,b.booking_no

```

Booking Grid View

```
SELECT S.salesperson_name          AS [Salesperson],
       o.loginname                 AS [Original Operator],
       B.booking_no                AS [Booking #],
       B.organizationv6            AS [Company],
       CASE
         WHEN B.booking_type_v32 = 0 THEN 'Rental'
         WHEN B.booking_type_v32 = 1 THEN 'Sub Hire'
         WHEN B.booking_type_v32 = 2 THEN 'Quote'
         WHEN B.booking_type_v32 = 3 THEN 'Rental Quote Monthly'
         WHEN B.booking_type_v32 = 4 THEN 'Rental Monthly'
         WHEN B.booking_type_v32 = 5 THEN 'Sub Rental'
         WHEN B.booking_type_v32 = 6 THEN 'Sale'
         WHEN B.booking_type_v32 = 7 THEN 'Sales Quote'
         WHEN B.booking_type_v32 = 11 THEN 'Location Transfer'
         WHEN B.booking_type_v32 = 13 THEN 'Sundry Transfer'
         WHEN B.booking_type_v32 = 14 THEN 'Rental 7'
         WHEN B.booking_type_v32 = 15 THEN 'Rental 14'
         WHEN B.booking_type_v32 = 16 THEN 'Rental 28'
         WHEN B.booking_type_v32 = 17 THEN 'Rental Quarterly'
         WHEN B.booking_type_v32 = 18 THEN 'Rental Quote 7'
         WHEN B.booking_type_v32 = 19 THEN 'rental_quote_14'
         WHEN B.booking_type_v32 = 20 THEN 'Rental Quote 28'
         WHEN B.booking_type_v32 = 21 THEN 'Rental Quote Quarterly'
         ELSE ''
       END                          AS Type,
       CASE
         WHEN B.status = 0 THEN 'Booked'
         WHEN B.status = 1 THEN 'Checked Out'
         WHEN B.status = 2 THEN 'Returned'
         ELSE ''
       END                          AS State,
       B.showname                  AS [Showname],
       B.invoiced                  AS [Invoiced],
       B.invoice_no                AS [Invoice #],
       Cast(b.price_quoted AS DECIMAL(9, 2)) AS [Price Quoted],
       B.contact_namev6            AS [Contact Name],
       B.ddate                     AS [Out Date],
       B.rdate                     AS [Return Date],
       V.venue_name                AS [Venue],
       CASE
         WHEN B.bookingprogressstatus = 0 THEN 'Quote'
         WHEN B.bookingprogressstatus = 1 THEN 'Light Pencil'
         WHEN B.bookingprogressstatus = 2 THEN 'Heavy Pencil'
         WHEN B.bookingprogressstatus = 3 THEN 'Confirmed'
         ELSE 'Cancelled'
       END                          AS [Progress Status]
FROM   tblbookings B
       LEFT OUTER JOIN tblvenues V
         ON B.venueid = V.id
       LEFT OUTER JOIN tblsalesper S
         ON B.salesperson = S.salesperson_code
       LEFT OUTER JOIN tbloperators O
         ON o.id = b.operatorsid
```

Bookings Entered Today

```

SELECT B.booking_no AS Code,
       B.organizationv6 AS Company,
       B.ddate AS [Out Date],
       D.div_name AS Division,
       CASE
           WHEN B.bookingprogressstatus = 0 THEN 'Quote'
           WHEN B.bookingprogressstatus = 1 THEN 'Light Pencil'
           WHEN B.bookingprogressstatus = 2 THEN 'Heavy Pencil'
           WHEN B.bookingprogressstatus = 3 THEN 'Confirmed'
           ELSE 'Cancelled'
       END AS [Progress Status],
       B.showname AS [Show Name],
       V.venue AS Venue,
       B.price_quoted AS Revenue,
       CASE
           WHEN (
               B.booking_type_v32 = 6
               OR B.booking_type_v32 = 7) THEN Cast(B.sales_discount_rate AS DECIMAL (6,2))
           ELSE Cast(B.discount_rate AS DECIMAL(6,2))
       END AS [Disc %],
       CASE
           WHEN B.booking_type_v32 = 6 THEN Cast(((b.sales_amount - B.sales_undisc_amount) * (b.sales_discount_rate /100))
           AS DECIMAL (9,2))
           ELSE Cast(((b.hire_price - B.un_disc_amount) * (b.discount_rate/100)) AS DECIMAL(9,2))
       END AS [Discount Amt],
       O.firstname AS [Operator First],
       O.lastname AS [Operator Last],
       B.salesperson,
       B.entrydate AS [Booking Created],
       B.condate AS [Confirmed Date],
       B.inv_date AS [Invoiced Date],
       CASE
           WHEN B.booking_type_v32 = 0 THEN 'Rental'
           WHEN B.booking_type_v32 = 1 THEN 'Sub Hire'
           WHEN B.booking_type_v32 = 2 THEN 'Quote'
           WHEN B.booking_type_v32 = 3 THEN 'Rental Quote Monthly'
           WHEN B.booking_type_v32 = 4 THEN 'Rental Monthly'
           WHEN B.booking_type_v32 = 5 THEN 'Sub Rental'
           WHEN B.booking_type_v32 = 6 THEN 'Sale'
           WHEN B.booking_type_v32 = 7 THEN 'Sales Quote'
           WHEN B.booking_type_v32 = 11 THEN 'Location Transfer'
           WHEN B.booking_type_v32 = 13 THEN 'Sundry Transfer'
           WHEN B.booking_type_v32 = 14 THEN 'Rental 7'
           WHEN B.booking_type_v32 = 15 THEN 'Rental 14'
           WHEN B.booking_type_v32 = 16 THEN 'Rental 28'
           WHEN B.booking_type_v32 = 17 THEN 'Rental Quarterly'
           WHEN B.booking_type_v32 = 18 THEN 'Rental Quote 7'
           WHEN B.booking_type_v32 = 19 THEN 'rental_quote_14'
           WHEN B.booking_type_v32 = 20 THEN 'Rental Quote 28'
           WHEN B.booking_type_v32 = 21 THEN 'Rental Quote Quarterly'
           ELSE ''
       END AS Type,
       CASE
           WHEN B.status = 0 THEN 'Booked'
           WHEN B.status = 1 THEN 'Checked Out'
           WHEN B.status = 2 THEN 'Returned'
           ELSE ''
       END AS State,
       B.invoiced,
       B.invoice_no
FROM dbo.tblbookings AS B
LEFT OUTER JOIN dbo.tbldivlist AS D

```

```

ON      B.division = D.div_number
LEFT OUTER JOIN dbo.tbloperators AS O
ON      B.operatorsid = O.id
LEFT OUTER JOIN dbo.tblvenues AS V
ON      B.venueid = V.id
WHERE CONVERT(VARCHAR(12),b.entrydate, 112) >= CONVERT(VARCHAR(12),Getdate(), 112)

```

Bookings Entered Within the Last 7 Days

```

Shows bookings entered IN the last 7 daysSELECT      B.booking_no   AS Code,
      B.organizationv6 AS Company,
      B.ddate         AS [Out Date],
      D.div_name     AS Division,
      CASE
        WHEN B.bookingprogressstatus = 0 THEN 'Quote'
        WHEN B.bookingprogressstatus = 1 THEN 'Light Pencil'
        WHEN B.bookingprogressstatus = 2 THEN 'Heavy Pencil'
        WHEN B.bookingprogressstatus = 3 THEN 'Confirmed'
        ELSE 'Cancelled'
      END         AS [Progress Status],
      B.showname     AS [Show Name],
      V.venueid     AS Venue,
      B.price_quoted AS Revenue,
      CASE
        WHEN (
          B.booking_type_v32 = 6
          OR      B.booking_type_v32 = 7) THEN Cast(B.sales_discount_rate AS DECIMAL (6,2))
        ELSE Cast(B.discount_rate AS DECIMAL(6,2))
      END AS [Disc %],
      CASE
        WHEN B.booking_type_v32 = 6 THEN Cast(((b.sales_amount - B.sales_undisc_amount) * (b.sales_discount_rate /100)) AS DECIMAL (9,2))
        ELSE Cast(((b.hire_price - B.un_disc_amount) * (b.discount_rate/100)) AS DECIMAL(9,2))
      END AS [Discount Amt],
      O.firstname AS [Operator First],
      O.lastname AS [Operator Last],
      B.salesperson,
      B.entrydate AS [Booking Created],
      B.condate AS [Confirmed Date],
      B.inv_date AS [Invoiced Date],
      CASE
        WHEN B.booking_type_v32 = 0 THEN 'Rental'
        WHEN B.booking_type_v32 = 1 THEN 'Sub Hire'
        WHEN B.booking_type_v32 = 2 THEN 'Quote'
        WHEN B.booking_type_v32 = 3 THEN 'Rental Quote Monthly'
        WHEN B.booking_type_v32 = 4 THEN 'Rental Monthly'
        WHEN B.booking_type_v32 = 5 THEN 'Sub Rental'
        WHEN B.booking_type_v32 = 6 THEN 'Sale'
        WHEN B.booking_type_v32 = 7 THEN 'Sales Quote'
        WHEN B.booking_type_v32 = 11 THEN 'Location Transfer'
        WHEN B.booking_type_v32 = 13 THEN 'Sundry Transfer'
        WHEN B.booking_type_v32 = 14 THEN 'Rental 7'
        WHEN B.booking_type_v32 = 15 THEN 'Rental 14'
        WHEN B.booking_type_v32 = 16 THEN 'Rental 28'
        WHEN B.booking_type_v32 = 17 THEN 'Rental Quarterly'
        WHEN B.booking_type_v32 = 18 THEN 'Rental Quote 7'
        WHEN B.booking_type_v32 = 19 THEN 'rental_quote_14'
        WHEN B.booking_type_v32 = 20 THEN 'Rental Quote 28'
        WHEN B.booking_type_v32 = 21 THEN 'Rental Quote Quarterly'

```

```

        ELSE ''
    END AS Type,
    CASE
        WHEN B.status = 0 THEN 'Booked'
        WHEN B.status = 1 THEN 'Checked Out'
        WHEN B.status = 2 THEN 'Returned'
        ELSE ''
    END AS State,
    B.invoiced,
    B.invoice_no
FROM    dbo.tblbookings AS B
LEFT OUTER JOIN dbo.tbldivlist AS D
ON      B.division = D.div_number
LEFT OUTER JOIN dbo.tbloperators AS O
ON      B.operatorsid = O.id
LEFT OUTER JOIN dbo.tblvenues AS V
ON      B.venueid = V.id
WHERE   CONVERT(VARCHAR(12),b.entrydate, 112) >= CONVERT(VARCHAR(12),Getdate()-7, 112)

```

Inventory Packages and Macros

Provides a list of all packages and macros in the database
including their components, accessories and alternatives

```

WITH CTE AS (
SELECT
    ISNULL(I.groupfld, '') AS ParentGroup,
    ISNULL(PG.group_descv6, '') AS ParentGroupDesc,
    ISNULL(I.category, '') AS ParentCategory,
    ISNULL(PC.cat_descv6, '') AS ParentCategoryDesc,
    ISNULL(I.subcategory, '') AS ParentSubCategory,
    ISNULL(PSC.cat_descv6, '') AS ParentSubCategoryDesc,
    ISNULL(B.parent_code, '') AS ParentCode,
    ISNULL(I.descriptionv6, '') AS ParentDescription,

    ISNULL(
        CASE I.product_config
            WHEN 0 THEN 'Product'
            WHEN 1 THEN 'PACKAGE'
            WHEN 2 THEN 'MACRO'
        END, ''
    ) AS ParentConfig,

```

```

ISNULL(I2.groupfld,'') AS ChildGroup,
ISNULL(CG.group_descv6,'') AS ChildGroupDesc,
ISNULL(I2.category,'') AS ChildCategory,
ISNULL(CC.cat_descv6,'') AS ChildCategoryDesc,
ISNULL(I2.subcategory,'') AS ChildSubCategory,
ISNULL(CSC.cat_descv6,'') AS ChildSubCategoryDesc,
ISNULL(B.product_code,'') AS ChildCode,
ISNULL(I2.descriptionv6,'') AS ChildDescription,

CASE B.variable_part

WHEN 0 THEN 'Component'

WHEN 1 THEN 'Accessory'

WHEN 2 THEN 'Alternative'

END AS ChildDefinedAs,

B.SelectComp AS ChildSelectComp,

ISNULL(CAST(B.qty_v5 AS varchar(20)),'') AS ChildQty,

ISNULL(
    CAST(I2.product_config AS varchar(10)),
    ''
) AS ChildConfig,

ROW_NUMBER() OVER (
    PARTITION BY B.parent_code
    ORDER BY B.sub_seq_no
) AS rn

FROM tblbill B

LEFT JOIN tblinvmas I ON I.product_code = B.parent_code

LEFT JOIN tblgroup PG ON PG.group_code = I.groupfld

LEFT JOIN tblcategory PC ON PC.category_code = I.category

LEFT JOIN tblcategory PSC ON PSC.category_code = I.subcategory

LEFT JOIN tblinvmas I2 ON I2.product_code = B.product_code

LEFT JOIN tblgroup CG ON CG.group_code = I2.groupfld

LEFT JOIN tblcategory CC ON CC.category_code = I2.category

```

```

LEFT JOIN tblcategory CSC ON CSC.category_code = I2.subcategory

WHERE I.product_type_v41 = 0

AND I.product_config IN (1,2)

)

SELECT

CASE WHEN rn = 1 THEN ParentGroup ELSE '' END      AS [Parent Group],
CASE WHEN rn = 1 THEN ParentGroupDesc ELSE '' END  AS [Parent Group Description],
CASE WHEN rn = 1 THEN ParentCategory ELSE '' END   AS [Parent Category],
CASE WHEN rn = 1 THEN ParentCategoryDesc ELSE '' END AS [Parent Category Description],
CASE WHEN rn = 1 THEN ParentSubCategory ELSE '' END AS [Parent Sub Category],
CASE WHEN rn = 1 THEN ParentSubCategoryDesc ELSE '' END AS [Parent Sub Category Description],
CASE WHEN rn = 1 THEN ParentCode ELSE '' END      AS [Parent Code],
CASE WHEN rn = 1 THEN ParentDescription ELSE '' END AS [Parent Description],
CASE WHEN rn = 1 THEN ParentConfig ELSE '' END    AS [Parent Config],

ChildGroup      AS [Child Group],
ChildGroupDesc  AS [Child Group Description],
ChildCategory   AS [Child Category],
ChildCategoryDesc AS [Child Category Description],
ChildSubCategory AS [Child Sub Category],
ChildSubCategoryDesc AS [Child Sub Category Description],
ChildCode       AS [Child Code],
ChildDefinedAs  AS [Type],
ChildSelectComp AS [Optional],
ChildDescription AS [Child Description],
ChildQty        AS [Child Qty],
ChildConfig     AS [Product Config]

FROM CTE

ORDER BY ParentGroup, ParentCategory, ParentSubCategory, ParentCode;

```

Inventory Packages

Provides a list of all packages in the database
including their components, accessories and alternatives

```

WITH CTE AS (
    SELECT
        ISNULL(B.parent_code,'') AS ParentCode,
        ISNULL(I.descriptionv6,'') AS ParentDescription,
        ISNULL(
            CASE I.product_config
                WHEN 0 THEN 'Product'
                WHEN 1 THEN 'PACKAGE'
                WHEN 2 THEN 'MACRO'
            END, ''
        ) AS ParentConfig,
        ISNULL(B.product_code,'') AS ChildCode,
        ISNULL(I2.descriptionv6,'') AS ChildDescription,
        CASE B.variable_part
            WHEN 0 THEN '0-Component'
            WHEN 1 THEN '1-Accessory'
            WHEN 2 THEN '2-Alternative'
        END AS ChildDefinedAs,
        B.SelectComp AS ChildSelectComp,
        ISNULL(CAST(B.qty_v5 AS varchar(20)), '') AS ChildQty,
        ISNULL(
            CAST(I2.product_config AS varchar(10)),
            ''
        ) AS ChildConfig,
        ROW_NUMBER() OVER (
            PARTITION BY B.parent_code
            ORDER BY B.sub_seq_no
        ) AS rn
    FROM tblbill B
    LEFT JOIN tblinvmas I ON I.product_code = B.parent_code
    LEFT JOIN tblinvmas I2 ON I2.product_code = B.product_code

```

```

WHERE I.product_type_v41 = 0

AND I.product_config IN (1)

)

SELECT

CASE WHEN rn = 1 THEN ParentCode ELSE '' END      AS [Package],

CASE WHEN rn = 1 THEN ParentDescription ELSE '' END  AS [Package Description],

ChildCode      AS [Product],

ChildDefinedAs  AS [Type],

ChildSelectComp AS [Optional],

ChildDescription AS [Product Description],

ChildQty       AS [Product Qty]

FROM CTE

ORDER BY ParentCode,ChildDefinedAs;

```

Inventory Macros

Provides a list of all macros in the database
including their components, accessories and alternatives

```

WITH CTE AS (

SELECT

ISNULL(B.parent_code,'')      AS ParentCode,

ISNULL(I.descriptionv6,'')    AS ParentDescription,

ISNULL(

CASE I.product_config

WHEN 0 THEN 'Product'

WHEN 1 THEN 'PACKAGE'

WHEN 2 THEN 'MACRO'

END, ''

) AS ParentConfig,

ISNULL(B.product_code,'')     AS ChildCode,

ISNULL(I2.descriptionv6,'')   AS ChildDescription,

CASE B.variable_part

WHEN 0 THEN '0-Component'

```

```

WHEN 1 THEN '1-Accessory'

WHEN 2 THEN '2-Alternative'

END AS ChildDefinedAs,

B.SelectComp          AS ChildSelectComp,

ISNULL(CAST(B.qty_v5 AS varchar(20)), '') AS ChildQty,

ISNULL(

    CAST(I2.product_config AS varchar(10)),

    ''

) AS ChildConfig,

ROW_NUMBER() OVER (

    PARTITION BY B.parent_code

    ORDER BY B.sub_seq_no

) AS rn

FROM tblbill B

LEFT JOIN tblinvmas I ON I.product_code = B.parent_code

LEFT JOIN tblinvmas I2 ON I2.product_code = B.product_code

WHERE I.product_type_v41 = 0

AND I.product_config IN (2)

)

SELECT

CASE WHEN rn = 1 THEN ParentCode ELSE '' END      AS [Macro],

CASE WHEN rn = 1 THEN ParentDescription ELSE '' END  AS [Macro Description],

ChildCode      AS [Product],

ChildDefinedAs AS [Type],

ChildSelectComp AS [Optional],

ChildDescription AS [Product Description],

ChildQty      AS [Product Qty]

FROM CTE

ORDER BY ParentCode,ChildDefinedAs;

```

