




# POS Commands Reference Guide

## Contact Information

Retail Management Hero  
454 West Napa Street, Unit B  
Sonoma, CA 95476  
[documentation@rhmpos.com](mailto:documentation@rhmpos.com)

For more information or support, contact your RMH  
Partner.



## Contents

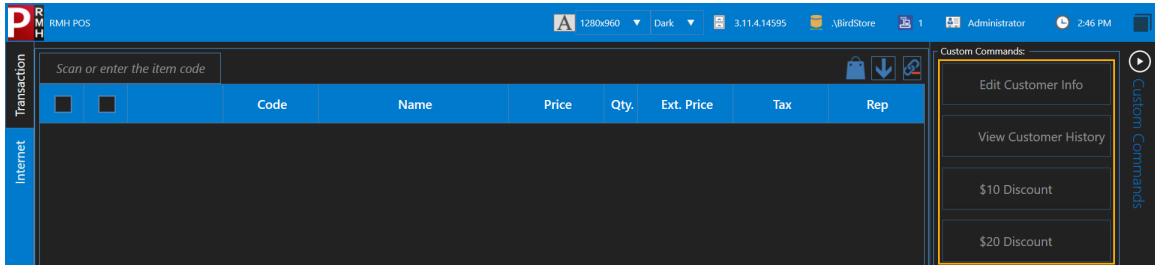
---

<b>Setting up custom POS buttons</b> .....	<b>1</b>
<b>Setting up custom POS task pads</b> .....	<b>2</b>
<b>Commands reference guide</b> .....	<b>5</b>
Batch Commands .....	5
Customer Commands .....	8
Discount Commands .....	10
Cash Drawer Commands .....	24
Item Commands .....	25
Item Tax Commands .....	30
Security Commands .....	31
Order Commands .....	31
Reason Code Commands .....	34
Receipt Printing Commands .....	36
Receipt Message Commands .....	37
Sales Representative Commands .....	38
Tendering Commands .....	40
Time Clock Command .....	42
Transaction Commands .....	42
Web Browser Command .....	46
Custom Action Command .....	47

---

## Setting up custom POS buttons

Stores have the option to set up custom buttons in POS for common tasks, such as applying dollar or percent off discounts or tendering common cash denominations such as 20s, 50s, or 100s. Custom POS buttons display in the **Custom Commands** pane to the right of the POS window:



**Note:** Custom POS buttons display in the Custom Commands pane only if there is no task pad assigned to the register. If there is a task pad assigned to the register or user, the task pad displays in the Custom Commands pane in POS instead of the custom buttons. To save space in the POS screen, click the arrow to hide the Custom Commands pane when it is not in use.

### To configure custom buttons for POS:

1. Identify the commands that contain the functionality you want to program into the button. Refer to the [Commands reference guide](#) for more information.
2. In Store Manager, click **Setup**.
3. Expand **Hardware**.
4. Click **Custom POS Buttons**.
5. Click **New**.

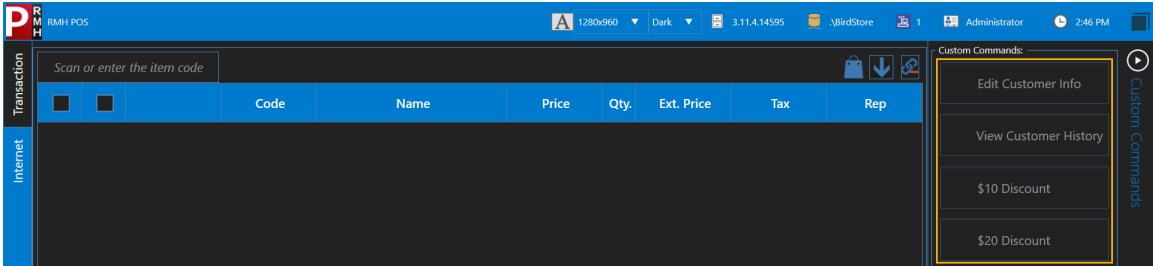
- 
6. For **Number**, enter a number representing the order in which the button should display in the **Custom Commands** pane, with 1 displaying at the top of the pane.
  7. From **Style**, select **Internal Command**.
  8. For **Caption**, enter the text that you would like to display on the button label, e.g., \$10 Discount.
  9. If desired, click **Load Image**, select an image to display on the button to the left of the button label, and click **Open**.
  10. In the **Command** field, enter the command (or chain of commands) and any relevant command parameters.

Starting with release 3.11.0 you can program Task Pads or Custom Buttons to execute a chain of commands when the button is clicked by adding a semicolon (;) between commands. For example, you might create a chain of commands to discount a transaction (command 1) and then automatically select a discount reason code (command 2). There is no limit to the number of commands in a chain. However, the commands should be chained together to follow a logical transaction process.

11. If desired, enter a **Description** of the command. This description does not display in POS. It is for reference only within Store Manager.
12. Click **Save And Close**.

## Setting up custom POS task pads

Stores have the option to set up task pads with custom buttons in POS for common tasks, such as looking up items. Task pad buttons display in the **Custom Commands** pane to the right of the POS window:



**Note:** It is also possible to configure custom POS buttons that display in the Custom Commands pane. Custom POS buttons only display if there is no task pad assigned to the register or user. If there is a task pad assigned to the register, the task pad displays in the Custom Commands pane instead of the custom POS buttons.

#### To configure task pads for POS:

1. Identify the commands that contain the functionality you want to program into the task pad buttons. Refer to the [Commands reference guide](#) for more information.
2. In Store Manager, click **Setup**.
3. Expand **Hardware**.
4. Click **POS Task Pads**.
5. Click **New**.
6. Enter a **Name** for the task pad.
7. From **Layout**, select a button layout for the task pad.
8. Click a button in the **Preview**.
9. From **Button Type**, select one of the following:

- 
- **Command:** Select this option if you want to configure the button to execute a command or chain of commands.
  - **Task Pad:** Select this option if you want to configure the button to launch a task pad. For example, if you run a convenience store and you use touch-screen monitors in POS lanes, you might want to set up task pads for different types of items like soft drinks, snacks, sandwiches, milk, and so on to make it easier to add items to transactions.
10. If you selected **Command**, in the **Command** field, enter the command (or chain of commands) and any relevant command parameters.

Starting with release 3.11.0 you can program Task Pads or Custom Buttons to execute a chain of commands when the button is clicked by adding a semicolon (;) between commands. For example, you might create a chain of commands to discount a transaction (command 1) and then automatically select a discount reason code (command 2). There is no limit to the number of commands in a chain. However, the commands should be chained together to follow a logical transaction process.

11. If you selected **Task Pad**, click the **Browse** icon next to **Task Pad Name**, select the task pad that you want to launch from the button, and click **OK**.
  12. If desired, select a custom color for the button. To do this, click the **Browse** icon next to **Button Color**, select the button color, and click **OK**.
  13. For **Caption**, enter the text that you would like to display on the button label, e.g., \$10 Discount.
  14. If desired, select a custom color for the font. To do this, click the **Browse** icon next to **Font Color**, select the button color, and click **OK**.
-

15. If desired, select an image to display on the button to the left of the button label. To do this, click the **Browse** icon next to **Image**, select the image, and click **Open**.
16. Click **Save And Close**.
17. Do one of the following:
  - Assign the task pad to a specific user; or
  - Assign the task pad to a specific register or user.

Refer to "Setting up a user account" and "Setting up registers" in the Store Manager User Guide for more information.

## Commands reference guide

The following commands can be used to program the functionality of both POS Task Pads and Custom POS Buttons. You must configure Task Pads and Custom Buttons in Store Manager.

Starting with release 3.11.0 you can program Task Pads or Custom Buttons to execute a chain of commands when the button is clicked by adding a semicolon (;) between commands. For example, you might create a chain of commands to discount a transaction (command 1) and then automatically select a discount reason code (command 2). There is no limit to the number of commands in a chain. However, the commands should be chained together to follow a logical transaction process.

## Batch Commands

Use batch commands to perform various batch operations, such as entering opening and closing amounts, viewing the X report, printing X, Z, and ZZ reports, and performing a blind closeout.

Command	Description
Batch_CloseCommand	Closes the current batch. This is the same as a blind closeout. No parameters.
Batch_EnterClosingAmountsCommand	Invokes the <b>Closing Amounts</b> dialog. No parameters.
Batch_Enter-OpeningAmountsCommand	Invokes the <b>Opening Amounts</b> dialog. No parameters.
Batch_PerformBlindCloseoutCommand	Performs a blind closeout. No parameters.
Batch_ViewXReportCommand	Displays the X report. No parameters.

### Charge Commands

Use charge commands to add charges to POS transactions.

Command	Description
Charge_AddNoTemplateCommand	<p>Adds a charge without using a charge template (defined in Store Manager under <b>Setup   Inventory/Purchasing   Charges</b>).</p> <p>The command takes an optional parameter of the charge code and price, sep-</p>



Command	Description
	<p>parated by a pipe ( ): {Charge Code}  {Price}.</p> <p>If a parameter is not provided, the command invokes the <b>Charge Lookup</b> dialog and prompts for the price.</p> <p><b>Example:</b></p> <p>Adds charge code and price:</p> <pre>Charge_AddNoTemplateCommand "CH 15.10"</pre>
Charge_AddCommand	<p>Adds a charge to the transaction using a charge template (defined in Store Manager under <b>Setup   Inventory/Purchasing   Charges</b>).</p> <p>The command takes an optional parameter of the charge code and price, separated by a pipe ( ): {Charge Code}  {Price}.</p> <p>If a parameter is not provided, the command invokes the <b>Charge Lookup</b> dialog and prompts for the price.</p> <p><b>Example:</b></p>

Command	Description
	<p>Adds charge code and price:</p> <pre>Charge_AddCommand "CHARG1 15.10"</pre> <p>Adds charge code but prompts for the price:</p> <pre>Charge_AddCommand "CHARG1"</pre>

## Customer Commands

Use customer commands are used to perform various customer-related operations.

Command	Description
Customer_AddCommand	<p>Adds a customer to the transaction. Requires the mandatory parameter {Customer Account Number}.</p> <p><b>Example:</b></p> <pre>Customer_AddCommand "000005"</pre>
Customer_ClearCommand	<p>Clears the current customer from the transaction. No parameters.</p>
Customer_DisplayHistoryCommand	<p>Displays a dialog with the current customer's purchase history. No para-</p>

Command	Description
	meters.
Customer_EditCommand	Invokes the <b>Customer</b> dialog where you can edit the customer. No parameters.
Customer_LookupCommand	<p>Adds a customer to the transaction.</p> <p>The command takes an optional parameter of {Customer Account Number}.</p> <p>If a parameter is not provided, or the customer account number is not found, the command invokes the <b>Customer Lookup</b> dialog with a list of customers fitting the search criteria.</p> <p><b>Example:</b></p> <p>Adds a customer with account number "000005" to the transaction:</p> <pre>Customer_LookupCommand "000005"</pre> <p>Invokes the Customer Lookup dialog with the search criteria "Jon":</p> <pre>Customer_LookupCommand "Jon"</pre>

Command	Description
Customer_NewCommand	Invokes the <b>Create New Customer Wizard</b> dialog where you can create a new customer. No parameters.
Customer_PaymentOnAccountCommand	Invokes the <b>Account Payment</b> dialog for the selected accounts receivable customer. No parameters.
Customer_SelectShippingAddressCommand	Invokes the <b>Select Shipping Address</b> dialog where you can add a shipping address for the selected customer. No parameters.

## Discount Commands

Use discount commands to apply various discounts and price levels on the selected item, all items in the transaction, or the entire transaction.

Command	Description
Discount_InvokeDiscountMenuCommand	Invokes the <b>Discount Menu</b> dialog. No parameters.

The following commands apply a discount to **all items** that are currently in the transaction.

**Note:** Discounts are not applied to items that are added after an all items command is used.

Command	Description
DiscountAllItems_SetNoDiscountCommand	Removes any discount applied to all items currently in the transaction. No parameters.
DiscountAllItems_SetAmountFromCurrentPriceCommand	<p>Applies a dollar amount discount from the current price to all items currently in the transaction.</p> <p>The command takes an optional parameter {Discount Amount}. If a parameter is not provided, the command invokes the <b>Discount</b> dialog and prompts for the dollar amount.</p> <p>Starting with release 3.11.12, the command also takes an optional parameter {Reason Code}. If the {Reason Code} parameter is provided the reason code is applied automatically to the item or transaction. The {Discount Amount} and {Reason Code} must be separated by a pipe ( ).</p> <p><b>Example:</b></p> <p>Apply a discount of \$10 dollars to the current price for all items currently in the transaction:</p>

Command	Description
	<p>DiscountAllItems_SetAmountFromCurrentPriceCommand "10"</p> <p>Apply the RC01 reason code:</p> <p>DiscountAllItems_SetAmountFromCurrentPriceCommand "10 RC01"</p>
DiscountAllItems_SetPercentFromCurrentPriceCommand	<p>Applies a percentage discount from the current price to all items currently in the transaction.</p> <p>The command takes an optional parameter {Percentage Discount}. If a parameter is not provided, the command invokes the <b>Discount</b> dialog and prompts for the percentage.</p> <p>Starting with release 3.11.12, the command also takes an optional parameter {Reason Code}. If the {Reason Code} parameter is provided the reason code is applied automatically to the item or transaction. The {Percentage Discount} and {Reason Code} must be separated by a pipe ( ).</p>

Command	Description
	<p><b>Example:</b></p> <p>Apply a 10% discount to the current price for all items currently in the transaction:</p> <pre>DiscountAllItems_SetPercentFromCurrentPriceCommand "10"</pre> <p>Apply the RC01 reason code:</p> <pre>DiscountAllItems_SetPercentFromCurrentPriceCommand "10 RC01"</pre>
DiscountAllItems_SetSubtotalAmountCommand	<p>Sets the transaction subtotal amount by proportionally discounting the transaction items.</p> <p>The command takes an optional parameter {Target Subtotal Amount}. If a parameter is not provided, the command invokes the <b>Discount</b> dialog and prompts for the dollar amount.</p> <p>Starting with release 3.11.12, the command also takes an optional parameter {Reason Code}. If the {Reason Code}</p>

Command	Description
	<p>parameter is provided the reason code is applied automatically to the item or transaction. The {Target Subtotal Amount} and {Reason Code} must be separated by a pipe ( ).</p> <p><b>Example:</b></p> <p>Set the subtotal amount to \$20 by discounting all items proportionally:</p> <pre>DiscountAllItems_SetSubtotalAmountCommand "20"</pre> <p>Apply the RC01 reason code:</p> <pre>DiscountAllItems_SetSubtotalAmountCommand "20 RC01"</pre>
DiscountAllItems_SetPriceLevelACommand	Applies Price Level A to all items currently in the transaction. No parameters.
DiscountAllItems_SetPriceLevelBCommand	Applies Price Level B to all items currently in the transaction. No parameters.
DiscountAllItems_SetPriceLevelCCommand	Applies Price Level C to all items currently in the transaction. No parameters.
DiscountAllItems_SetPriceLevelDe-	Removes the price level from all items cur-



Command	Description
faultCommand	rently in the transaction. No parameters.

The following commands apply a discount to the **currently selected item**.

DiscountCurrentItem_SetAmountFromCurrentPriceCommand	<p>Applies a dollar amount discount to the current price for the currently selected item.</p> <p>The command takes an optional parameter {Discount Amount}. If a parameter is not provided, the command invokes the <b>Discount</b> dialog and prompts for the dollar amount.</p> <p>Starting with release 3.11.12, the command also takes an optional parameter {Reason Code}. If the {Reason Code} parameter is provided the reason code is applied automatically to the item or transaction. The {Discount Amount} and {Reason Code} must be separated by a pipe ( ).</p> <p><b>Example:</b></p> <p>Apply a discount of \$10 dollars to the current price for the currently selected item:</p>
--	---

Command	Description
	<p>DiscountCurrentItem_SetAmountFromCurrentPriceCommand "10"</p> <p>Apply the RC01 reason code:</p> <p>DiscountCurrentItem_SetAmountFromCurrentPriceCommand "10 RC01"</p>
DiscountCurrentItem_SetNoDiscountCommand	Removes any applied discount from the selected item. No parameters.
DiscountCurrentItem_SetPercentFromCurrentPriceCommand	<p>Applies a percentage discount to the current price for the currently selected item.</p> <p>The command takes an optional parameter {Discount Percentage}. If a parameter is not provided, the command invokes the <b>Discount</b> dialog.</p> <p>Starting with release 3.11.12, the command also takes an optional parameter {Reason Code}. If the {Reason Code} parameter is provided the reason code is applied automatically to the item or transaction. The {Discount Percentage} and {Reason Code} must be separated by a</p>

Command	Description
	<p>pipe ( ).</p> <p><b>Example:</b></p> <p>Apply a 10% discount to the current price for the currently selected item:</p> <pre>DiscountCurrentItem_SetPercentFromCurrentPriceCommand "10"</pre> <p>Apply the RC01 reason code:</p> <pre>DiscountCurrentItem_SetPercentFromCurrentPriceCommand "10 RC01"</pre>
DiscountCurrentItem_SetPercentFromRetailPriceCommand	<p>Applies a percentage discount to the retail price for currently selected item.</p> <p>The command takes an optional parameter {Discount Percentage}. If a parameter is not provided, the command invokes the <b>Discount</b> dialog.</p> <p>Starting with release 3.11.12, the command also takes an optional parameter {Reason Code}. If the {Reason Code} parameter is provided the reason code is</p>

Command	Description
	<p>applied automatically to the item or transaction. The {Discount Percentage} and {Reason Code} must be separated by a pipe ( ).</p> <p><b>Example:</b></p> <p>Apply a 10% discount to the retail price for the currently selected item:</p> <pre>DiscountCurrentItem_SetPercentFromRetailPriceCommand "10"</pre> <p>Apply the RC01 reason code:</p> <pre>DiscountCurrentItem_SetPercentFromRetailPriceCommand "10 RC01"</pre>
DiscountCurrentItem_SetProfitMarginCommand	<p>Sets the profit margin for the currently selected item.</p> <p>The command takes an optional parameter {Profit Margin Percentage}. If a parameter is not provided, the command invokes the <b>Discount</b> dialog.</p> <p>Starting with release 3.11.12, the com-</p>

Command	Description
	<p>mand also takes an optional parameter {Reason Code}. If the {Reason Code} parameter is provided the reason code is applied automatically to the item or transaction. The {Profit Margin Percentage} and {Reason Code} must be separated by a pipe ( ).</p> <p><b>Example:</b></p> <p>Sets a 10% profit margin for the currently selected item:</p> <pre>DiscountCurrentItem_ SetProfitMarginCommand "10"</pre> <p>Apply the RC01 reason code:</p> <pre>DiscountCurrentItem_ SetProfitMarginCommand "10 RC01"</pre>
DiscountCurrentItem_ SetPriceLevelACommand	Applies Price Level A to the currently selected item. No parameters.
DiscountCurrentItem_ SetPriceLevelBCommand	Applies Price Level B to the currently selected item. No parameters.

Command	Description
DiscountCurrentItem_ SetPriceLevelCCommand	Applies Price Level C to the currently selected item. No parameters.
DiscountCurrentItem_ SetPriceLevelDefaultCommand	Removes any Price Level from the currently selected item. No parameters.

Use following commands to apply a discount to **all items that are currently in the transaction or will be added later.**

DiscountTransaction_ SetNoDiscountCommand	Removes the transaction discount. No parameters.
DiscountTransaction_ SetFromMinimumPriceCommand	<p>Applies a percentage discount to the minimum price for the transaction.</p> <p>The command takes an optional parameter {Discount Percentage}. If a parameter is not provided, the command invokes the Discount dialog.</p> <p>Starting with release 3.11.12, the command also takes an optional parameter {Reason Code}. If the {Reason Code} parameter is provided the reason code is applied automatically to the item or transaction. The {Discount Percentage} and {Reason Code} must be separated by a pipe ( ).</p>

Command	Description
	<p><b>Example:</b></p> <p>Apply a 10% discount to the minimum price for the transaction:</p> <pre>DiscountTransaction_SetFromMinimumPriceCommand "10"</pre> <p>Apply the RC01 reason code:</p> <pre>DiscountTransaction_SetFromMinimumPriceCommand "10 RC01"</pre>
DiscountTransaction_SetFromRetailPriceCommand	<p>Applies a percentage discount to the retail price for the transaction.</p> <p>The command takes an optional parameter {Discount Percentage}. If a parameter is not provided, the command invokes the Discount dialog.</p> <p>Starting with release 3.11.12, the command also takes an optional parameter {Reason Code}. If the {Reason Code} parameter is provided the reason code is applied automatically to the item or transaction. The {Discount Percentage} and {Reason Code} must be separated by a</p>

Command	Description
	<p>pipe ( ).</p> <p><b>Example:</b></p> <p>Apply a 10% discount to the retail price for the transaction:</p> <pre>DiscountTransaction_SetFromRetailPriceCommand "10"</pre> <p>Apply the RC01 reason code:</p> <pre>DiscountTransaction_SetFromRetailPriceCommand "10 RC01"</pre>
DiscountTransaction_SetMarkupFromCostCommand	<p>Sets the transaction markup from cost.</p> <p>The command takes an optional parameter {Markup Percentage}. If a parameter is not provided, the command invokes the Discount dialog.</p> <p>Starting with release 3.11.12, the command also takes an optional parameter {Reason Code}. If the {Reason Code} parameter is provided the reason code is applied automatically to the item or transaction. The {Markup Percentage} and {Reason Code} must be separated by a</p>



Command	Description
	<p>pipe ( ).</p> <p><b>Example:</b></p> <pre>DiscountTransaction_ SetMarkupFromCostCommand "10"</pre> <p>Apply the RC01 reason code:</p> <pre>DiscountTransaction_ SetMarkupFromCostCommand "10 RC01"</pre>
DiscountTransaction_SetProfitMarginCommand	<p>Sets the transaction profit margin.</p> <p>The command takes an optional parameter {Profit Margin Percentage}. If a parameter is not provided, the command invokes the Discount dialog.</p> <p>Starting with release 3.11.12, the command also takes an optional parameter {Reason Code}. If the {Reason Code} parameter is provided the reason code is applied automatically to the item or transaction. The {Profit Margin Percentage} and {Reason Code} must be separated by a pipe ( ).</p>

Command	Description
	<p><b>Example:</b></p> <pre>DiscountTransaction_ SetProfitMarginCommand "10"</pre> <p>Apply the RC01 reason code:</p> <pre>DiscountTransaction_ SetProfitMarginCommand "10 RC01"</pre>
DiscountTransaction_ SetPriceLevelACommand	Sets the transaction price level to Price Level A. No parameters.
DiscountTransaction_ SetPriceLevelBCommand	Sets the transaction price level to Price Level B. No parameters.
DiscountTransaction_ SetPriceLevelCCommand	Sets the transaction price level to Price Level C. No parameters.
DiscountTransaction_ SetPriceLevelDefaultCommand	Removes the transaction price level. No parameters.

## Cash Drawer Commands

Use the cash drawer commands to perform cash drops and pay outs from the cash drawer.

Command	Description
Drawer_CashDropCommand	Initiates a cash drop and invokes the <b>Drop</b> dialog. No parameters.
Drawer_CashPayOutCommand	Performs a pay out and invokes the Pay Out dialog. No parameters.

## Item Commands

Use item commands to perform various item functions, such as adding an item to a transaction, deleting an item from a transaction, and so on.

Command	Description
Item_AddCommand	<p>Adds an item to a transaction.</p> <p>The command takes a parameter {Item Lookup Code}.</p> <p><b>Example:</b></p> <p>Adds an item with the item lookup code 12345678 to the transaction:</p> <pre>Item_AddCommand "12345678"</pre>
Item_AddCommentCommand	<p>Adds a comment to the currently selected item.</p> <p>The command takes an optional para-</p>

	<p>meter {Comment Text}.</p> <p>If a parameter is not provided, the command invokes the <b>Item Comment</b> dialog.</p> <p>Example:</p> <pre>Item_AddCommentCommand "My comment on this item"</pre>
Item_ClearAllCommand	<p>Clears all items in the transaction. No parameters.</p>
Item_DeleteCommand	<p>Deletes an item from the transaction.</p> <p>The command takes a parameter {Item Lookup Code}.</p> <p>If a parameter is not provided, the command deletes the selected item(s).</p> <p><b>Example:</b></p> <p>Deletes item 12345678 from the transaction, if found:</p> <pre>Item_DeleteCommand "12345678"</pre> <p>Deletes the currently selected item(s) from the transaction:</p>

	Item_DeleteCommand
Item_DuplicateCommand	<p>Duplicates an item in the transaction.</p> <p>The command takes a parameter {Item Lookup Code}.</p> <p>If a parameter is not provided, the command duplicates the selected item.</p> <p><b>Example:</b></p> <p>Duplicates item 12345678 in the transaction, if found:</p> <pre>Item_DuplicateCommand "12345678"</pre> <p>Duplicates the currently selected item in the transaction:</p> <pre>Item_DuplicateCommand</pre>
Item_InvokeltemActionsDialogCommand	Invokes the <b>Item Action</b> dialog for the selected item. No parameters.
Item_InvokeltemPriceCheckerDialogCommand	Invokes the <b>Item Price Checker</b> dialog in POS.

	<p><b>Note:</b> The Item Price Checker dialog can only be used to check prices for standard items. It cannot be used to check prices for matrix, lot matrix, or assembly items.</p> <p>The command takes an optional parameter {ON}.</p> <p>If the parameter is provided, the Item Price Checker dialog will display on net display if net display is enabled for the register. If the parameter is not provided, the dialog will not display on net display.</p> <p><b>Example:</b></p> <p>Displays the dialog on net display if net display is enabled for the register:</p> <pre>Item_InvokeItemPriceCheckerDialogCommand ON</pre>
Item_LookupCommand	<p>Invokes the <b>Item Lookup</b> dialog.</p> <p>The command takes an optional parameter {Search Criteria}, which may be a combination of item lookup code, descrip-</p>

	<p>tion etc.</p> <p>If a parameter is not provided, the command invokes the <b>Item Lookup</b> dialog without any search criteria.</p> <p><b>Example:</b></p> <p>Invokes the Item Lookup dialog with search criteria "123 My":</p> <pre>Item_LookupCommand "123 My"</pre>
Item_LookupDepCatCommand	<p>Initiates item department or category lookup.</p> <p>The command takes an optional parameter {Search Criteria} which may be a department or category name, or a combination of both.</p> <p>If both a department and category name are in search criteria, separate them with a forward slash (/).</p> <p><b>Example:</b></p> <p>Initiates item search for department or category "Camping":</p>

	<pre>Item_LookupDepCatCommand "Camp- ing"</pre>
	<p>Initiates item search by department and category "Clothing/Boys":</p>
	<pre>Item_LookupDepCatCommand "Cloth- ing/Boys"</pre>

## Item Tax Commands

Use item tax commands to change item taxes.

Command	Description
ItemTax_SetDestinationTaxCommand	Applies the destination tax based on the shipping address. No parameters.
ItemTax_SetNoItemTaxForTransactionCommand	Makes the transaction tax free. No parameters.
ItemTax_SetForCurrentItemCommand	Invokes the <b>Select Tax</b> dialog for the currently selected item. No parameters.
ItemTax_SetForTransactionCommand	Invokes the <b>Select Tax</b> for the transaction. No parameters.



Command	Description
ItemTax_ToggleForItemCommand	Toggles item tax on/off for the selected item(s). No parameters.
ItemTax_ToggleForTransactionCommand	Toggles the transaction item tax on/off. No parameters.

## Security Commands

Use the security command to invoke perform a cashier override.

Command	Description
Security_ChangeUserCommand	Logs out the current user and Invokes the Login dialog. Takes no parameter.

## Order Commands

Use the order commands for functionality related to work orders, back orders, lay-aways, and quotes.

Command	Description
Order_AddCommentCommand	<p>Adds a comment to an order.</p> <p>The command takes an optional parameter {Comment Text}.</p> <p>If a parameter is not provided, the command invokes the <b>Order Comment</b> dia-</p>

Command	Description
	<p>log.</p> <p><b>Example:</b></p> <pre>Order_AddCommentCommand "Order comment"</pre>
Order_DepositCommand	<p>Invokes the <b>Enter Deposit</b> dialog. No parameters.</p>
Order_OverrideDepositPercentageCommand	<p>Overrides the order deposit with a percentage of the order total.</p> <p>The command takes a parameter {Deposit Percentage}.</p> <p><b>Usage:</b></p> <p>Override the order deposit with 10% of order total:</p> <pre>Order_OverrideDepositPercentageCommand "10"</pre>
Order_OverrideDepositAmountCommand	<p>Overrides the order deposit with an amount.</p> <p>The command takes a parameter {Deposit</p>

Command	Description
	<p>Amount}.</p> <p><b>Example:</b></p> <p>Override the order deposit with an amount of \$100:</p> <pre>Order_OverrideDepositAmountCommand "100"</pre>
Order_ChangDueDateCommand	Invokes the <b>Order Due Date</b> dialog. No parameters.
WorkOrder_RecallCommand	Invokes the <b>Recall Work Order</b> dialog where you can select a work order. No parameters.
Quote_RecallCommand	Invokes the <b>Recall Quote</b> dialog where you can select a quote. No parameters.
BackOrder_RecallCommand	Invokes the <b>Recall Back Order</b> dialog where you can select a back order. No parameters.
Layaway_RecallCommand	Invokes the <b>Recall Layaway</b> dialog where you can select a layaway. No parameters.

Command	Description
Layaway_PaymentCommand	Initiates the payment on layaway process. No parameters.

## Reason Code Commands

Use the reason code commands to apply reason codes to items and transactions.

Command	Description
ReasonCode_ SetReasonCodeForCur- rentItemCommand	<p>Applies a reason code to the currently selected item.</p> <p>The command takes a parameter {Reason Code}.</p> <p><b>Example:</b></p> <p>Apply the RMI01 reason code to the selected item:</p> <pre>ReasonCode_SetReasonCodeForCur- rentItemCommand "RMI01"</pre>
ReasonCode_ SetReasonCodeForItemCommand	<p>Applies a reason code to an item, using the transaction item key and the reason code.</p> <p><b>Example:</b></p> <p>Apply the RC01 reason code to an item</p>

Command	Description
	<p>with the item key "2":</p> <pre>ReasonCode_ SetReasonCodeForItemCommand "2 RC01"</pre>
<pre>ReasonCode_ SetReasonCodeForTrans- actionCommand</pre>	<p>Applies a reason code to the transaction.</p> <p>The command takes a parameter {Reason Code}.</p> <p><b>Example:</b></p> <p>Apply the RMI01 reason code to the transaction:</p> <pre>ReasonCode_ SetReasonCodeForTrans- actionCommand "RMI01"</pre>
<pre>ReasonCode_ SetReasonCodeForAllItem- sCommand</pre>	<p>Applies a reason code to all items in the transaction.</p> <p>The command takes a parameter {Reason Code}.</p> <p><b>Example:</b></p> <p>Apply the RMI01 reason code to all items</p>

Command	Description
	<p>in the transaction:</p> <pre>ReasonCode_ SetReasonCodeForAllItems Command "RMI01"</pre>

## Receipt Printing Commands

Use these commands for receipt printing.

Command	Description
Receipt_PrintXReportCommand	Prints the X report. No parameters.
Receipt_PrintZReportCommand	Prints the Z report. No parameters.
Receipt_PrintZZReportCommand	Prints the ZZ report. No parameters.
Receipt_ReprintCommand	Reprints the last receipt. No parameters.
Receipt_ReprintGiftCommand	Reprints the last receipt as a gift receipt. The receipt template must be configured to contain a gift receipt section. No parameters.
Receipt_ReprintWithTemplateCommand	Initiates the process of reprinting the last receipt using a different template. No parameters.

Command	Description
Receipt_PreviewCommand	<p>Previews the receipt.</p> <p>The command takes a parameter {"Printer1"} or {"Printer2"} to select which printer to preview. The specified printer to be configured to use a receipt template.</p> <p><b>Example:</b></p> <p>Preview the receipt for Printer1:</p> <pre>Receipt_PreviewCommand "Printer1"</pre>

## Receipt Message Commands

Use receipt message commands to set or clear custom session variables used in the receipt template to print custom messages. Requires a customized receipt template which takes advantage of these functions.

Command	Description
ReceiptMessage_RemoveCustomMessageCommand	<p>Clears custom message from a receipt session variable.</p> <p>The command takes a parameter {Session Variable Number.}</p>

Command	Description
	<p><b>Example:</b></p> <p>Clears session variable1:</p> <pre>ReceiptMessage_RemoveCustomMessageCommand "1"</pre>
ReceiptMessage_SetCustomMessageCommand	<p>Sets a custom message to a session variable for printing on the receipt.</p> <p>The command takes the parameters {Session Variable Number} and a {Value} separated by a pipe ( ).</p> <p><b>Example:</b></p> <p>Sets session variable 1 to "My Message":</p> <pre>ReceiptMessage_SetCustomMessageCommand "1 My message"</pre>

## Sales Representative Commands

Use sales representative commands to set (or clear) sales representatives for items or transactions.



Command	Description
SalesRep_SetForCurrentItemCommand	<p>Applies a sales representative to the currently selected item.</p> <p>The command takes an optional parameter {Sales Representative Number}.</p> <p>If a parameter is not provided, or the number is not found, the command invokes the <b>Sales Rep Lookup</b> dialog where you can select a sales representative.</p> <p><b>Example:</b></p> <p>Apply the sales representative "001" to the currently selected item:</p> <pre>SalesRep_SetForCurrentItemCommand "001"</pre>
SalesRep_SetForTransactionCommand	<p>Applies a sales representative to the transaction.</p> <p>The command takes an optional parameter {Sales Representative Number}.</p> <p>If a parameter is not provided, or the number is not found, the command invokes the <b>Sales Rep Lookup</b> dialog</p>

Command	Description
	<p>where you can select a sales representative.</p> <p><b>Example:</b></p> <p>Apply the sales representative "002" to the transaction:</p> <pre>SalesRep_SetForTransactionCommand "002"</pre>
SalesRep_ClearFromCurrentItemCommand	Removes the assigned sales representative from the currently selected item. No parameters.
SalesRep_ClearFromTransactionCommand	Removes the assigned sales representative from the transaction. No parameters.

## Tendering Commands

Use the tendering commands to initiate functions related to transaction tendering. Is used in the same way as Transaction\_InvokeTenderDialogCommand.

Command	Description
Tender_BeginCommand	Initiates the tendering process.
Tender_FixedAmountCommand	Tenders a specific tender type (by

Command	Description
	<p>tender code) with a specified amount.</p> <p>The command takes two parameters {Tender Code} and {Decimal Amount} separated by a pipe ( ).</p> <p><b>Example:</b></p> <p>Tender a transaction using tender type CSH with an amount \$10.25;</p> <pre>Tender_FixedAmountCommand "CSH 10.25"</pre> <p>Starting with release 3.11.20, this command executes the Loyalty program, if applicable, so customers can earn or redeem points on the transaction. The command also opens the Tender Dialog because, at the time the command is executed, the amount of the Loyalty discount is not known.</p>
Transaction_ExactTender-AmountCommand	<p>Tenders a specific tender type (by tender code) with the exact total amount.</p> <p>The command takes a parameter</p>

Command	Description
	<p>{TenderCode}.</p> <p><b>Example:</b></p> <p>Tenders a transaction for the exact total amount using tender type CSH:</p> <pre>Transaction_ExactTender-AmountCommand "CSH"</pre> <p>Starting with release 3.11.20, this command executes the Loyalty program, if applicable, so customers can earn or redeem points on the transaction.</p>

## Time Clock Command

Use this command to display the Time Clock screen.

Command	Description
TimeClock_InvokeDialogCommand	Displays the Time Clock screen. No parameters.

## Transaction Commands

Use these commands to manage the transaction flow.

Command	Description
Transaction_CancelTransactionCommand	<p> Cancels the transaction and controls whether the application is closed or not.</p> <p> The command takes the optional parameters {True} or {False} or blank.</p> <p><b>Example:</b></p> <p> Cancel the transaction and close the application:</p> <pre>Transaction_CancelTransactionCommand "True"</pre> <p> Or:</p> <pre>Transaction_CancelTransactionCommand</pre> <p> Cancel the transaction and do not close the application:</p> <pre>Transaction_CancelTransactionCommand "False"</pre>
Transaction_ChangeToLayawayCommand	<p> Changes the transaction to a layaway. No parameters.</p>
Transaction_ChangeToQuoteCom-	<p> Changes the transaction to quote. No</p>

Command	Description
mand	parameters.
Transaction_ChangeToWorkOrder- Command	Changes the transaction to work order. No parameters.
Transaction_AddCommentCommand	<p>Adds a comment to the transaction .</p> <p>The command takes an optional parameter {Comment Text}.</p> <p>If a parameter is not provided, the command invokes the <b>Transaction Comment</b> dialog.</p> <p><b>Example:</b></p> <pre>Transaction_AddCommentCommand "Comment text"</pre>
Transaction_SetReturnModeCommand	Changes the transaction to return transaction. No parameters.
Transaction_SetSalesModeCommand	Changes the transaction from return to sale. No parameters.
Transaction_SetOnHoldCommand	Puts the current transaction on hold. No parameters.
Transaction_InvokeTender-	Invokes the <b>Tender</b> dialog and initiates

Command	Description
DialogCommand	the tendering process. No parameters. Similar to Tender_BeginCommand.
Transaction_NetDisplayCommand	<p>Toggles net display on/off. Net display must be enabled in Store Manager.</p> <p>The command takes the parameters {ON} or {OFF}.</p> <p>If a parameter is not provided, the command toggles the net display on/off.</p> <p><b>Example:</b></p> <p>Turn on net display:</p> <pre>Transaction_NetDisplayCommand "ON"</pre> <p>Turn off net display:</p> <pre>Transaction_NetDisplayCommand "OFF"</pre> <p>Toggle net display on/off:</p> <pre>Transaction_NetDisplayCommand</pre>
Transaction_NoSaleCommand	Performs a no sale transaction. No parameters.

Command	Description
Transaction_RecallForReturnCommand	Initiates the recall for return process. No parameters.
Transaction_RecallForVoidCommand	Initiates the recall for void process. No parameters.
Transaction_RecallFromHoldCommand	Recalls a transaction from hold. No parameters.
Transaction_ToggleReturnModeCommand	Toggles the transaction return mode on/ff. No parameters.
Transaction_ViewJournalCommand	Displays the journal. No parameters.

## Web Browser Command

Use this command to display a URL in the built-in web browser.

Command	Description
WebBrowser_NavigateCommand	<p>Displays a URL (web page) in the built-in browser.</p> <p>The command takes a parameter {URL}.</p> <p><b>Example:</b></p> <pre>WebBrowser_NavigateCommand</pre>



Command	Description
	"http://www.retailhero.com"

## Custom Action Command

Use this command to enable custom functionality in a POS Extension using a custom button (or a task pad button).

Command	Description
CustomActionCommand	<p><b>Example:</b></p> <pre>CustomActionCommand "param1 param2 param3"</pre> <p>Essentially, this command fires up an event in the POS, and passes a parameter to it. Your extension will listen to that event, and when it is fired, check the parameter to see if it is the "own" parameter (which will be processed) or a "foreign" one (which will be ignored). The command/event is available equally to all extensions, so it is important for the extension to distinguish by looking at the parameter.</p> <p>Please review the demo project in the SDK:</p>

Command	Description
	<p>C:\Program Files (x86)\Retail Hero\RMH POS\RMH POS SDK\RMHDe-moExtension-Native\RMHDemoExtension</p> <p>(or the Telerik version of the same).</p> <p>Check the area "Custom Button Handling". The comment in the project reads:</p> <pre>// The custom buttons could use any string parameter (but only one per command - XAML allows only one command parameter)</pre> <pre>// Multiple parameters could be passed as a single string with separators, such as: "parameter1 parameter2 parameter3"</pre> <pre>// It is recommended to use GUIDs as parameter. That will make sure that different extensions don't mix-up handling</pre> <pre>// the same custom action event.</pre> <pre>// In the CustomPanel xaml, the Custom Action 1 and 2 buttons use guids as parameters</pre>

Command	Description
	<p>// Each custom button raises the same event, but passes a different parameter</p> <p>// That's how multiple extensions know which event is "their" and ignore all others.</p> <p>//Guid for Custom Action 1 button: DEDE0403-5C7A-4A84-BB75-F1DD0D914F4D</p> <p>//Guid for Custom Action 2 button: 71D09398-1A5C-4CB4-9738-1584B61483F1</p> <p>Although this explanation is a bit outdated (it talks about CustomPanel.xaml while since RMH version 3.2 we handle the custom buttons by creating them in Store Manager, not via the XAML modification), it still applies in principle no matter how the button is created.</p> <p>In fact, in your custom button you can use any "string" parameter with your command, but we recommend using generated GUIDs as that would prevent the possible extension "clash" if two or more</p>

Command	Description
	<p>extensions happen to use the same string parameter.</p> <p>See also p.19 in RMH_SDKProjects_v1_CL.pdf (in the POS SDK folder).</p>