iOrder: Product Scripts

A list of products scripts that are used commonly used for iOrder, with examples of how they can be used.

Note: more product scripts will be added as time permits.

KEY

Placeholder	Details	Example	More Info
X	limit to day of week	1-7	
XXX	limit to day abbreviation	MON or WED	
XXXXXX	limit to day expressions	WEEKDAYS or WEEKENDS	
XM	limit to day of month	16	16th day of the month
Y	limit to order source	2	
Z	limit to order type	3	
$	price override	5	
P	Product	P1001	
T	Matrix Target	T1357	
D	Department	D246	
CE	Combo Element	PIZZA or BREAD or 125DRINK	combo elements MUST be defined before they can be used
[ABC]	limit to price Structure	SPECIALPRICES	
DDMONYYYY	Date in DDMONYYYY format	01JAN2012	1st January 2012
DD-MON-YYYY	Date in DD-MON-YYYY format	01-JAN-2012	1st January 2012
TRADEPERIOD	Time period	BREAKFAST or LUNCH or DINNER	

PRICE

Overrides the POS price of the product.

Default use:

```
PRICE = $
```

Example - product price is $5 for all stores:

```
PRICE = 5
```

—

Struct use:

```
PRICE [ABC] = $
```

Example - product price is $5 for stores using price struct SPECIALPRICES:

```
PRICE [SPECIALPRICES] = 5
```

OLOPRICE

Overrides the POS price of the product when purchased ONLINE.

Default use:

```
OLOPRICE = $
```

Example - online product price is $5 for all stores:

```
OLOPRICE = 5
```

—

Struct use:

```
OLOPRICE [ABC] = $
```

Example - online product price is $5 for stores using price struct SPECIALPRICES:

```
OLOPRICE [SPECIALPRICES] = 5
```

PRICEDAY

Overrides the POS price of the product for the specified day.

Default use:

```
PRICEDAY X = $
```

Example - on Tuesday limit the price to $5 for all stores:

```
PRICEDAY 2 = 5
```

—

Struct use:

```
PRICEDAY [ABC] X = $
```

Example - on Tuesday limit the price to $5 for stores using price struct SPECIALPRICES:

```
PRICEDAY [SPECIALPRICES] 2 = 5
```

PRICEDAYSRC

Overrides the POS price of the product for the specified day and order source.

Default use:

```
PRICEDAYSRC X,Y = $
```

Example - on Tuesday if order source = 1 limit the price to $5 for all stores:

```
PRICEDAYSRC 2,1 = 5
```

—

Struct use:

```
PRICEDAYSRC [ABC] X,Y = $
```

Example - on Tuesday if order source = 1 limit the price to $5 for stores using price struct SPECIALPRICES:

```
PRICEDAYSRC [SPECIALPRICES] 2,1 = 5
```

PRICESOURCE

Overrides the POS price of the product for the specified order source.

Default use:

```
PRICESOURCE Y = $
```

Example - if order source = 1 limit the price to $5 for all stores:

```
PRICESOURCE 1 = 5
```

—

Struct use:

```
PRICESOURCE [ABC] Y = $
```

Example - if order source = 1 limit the price to $5 for stores using price struct SPECIALPRICES:

```
PRICESOURCE [SPECIALPRICES] 1 = 5
```

PRICETYPE

Overrides the POS price of the product for the specified order type.

Default use:

```
PRICETYPE Z = $
```

Example - if order type = 3 limit the price to $5 for all stores:

```
PRICETYPE 3 = 5
```

—

Struct use:

```
PRICETYPE [ABC] Z = $
```

Example - if order type = 3 limit the price to $5 for stores using price struct SPECIALPRICES:

```
PRICETYPE [SPECIALPRICES] 3 = 5
```

OTDAYPRICE

Overrides the POS price of the product for the specified order type and day.

Default use:

```
OTDAYPRICE Z,X = $
```

Example - on Tuesday if order type = 3 limit the price to $5 for all stores:

```
OTDAYPRICE 3,2 = 5
```

—

Struct use:

```
OTDAYPRICE [ABC] Z,X = $
```

Example - on Tuesday if order type = 3 limit the price to $5 for stores using price struct SPECIALPRICES:

```
OTDAYPRICE [SPECIALPRICES] 3,2 = 5
```

ONLINEDAYS

Sets the days when this product is available for online purchase.

Single day use:

```
ONLINEDAYS = X
```

Example - on Tuesday this product can be purchases for all stores:

```
ONLINEDAYS = 2
```

—

Multiple days use:

```
ONLINEDAYS = X,X,X
```

Example - on Tuesday, Thursday and Friday this product can be purchases for all stores:

```
ONLINEDAYS = 2,4,5
```

DAY RESTRICTIONS

Restricts the purchase of the product to the specified day(s) of the week.

Single day use:

```
DAY = XXX
```

Example - on Tuesday this product can be purchases for all stores:

```
DAY = TUE
```

—

Multiple days use:

```
DAY = XXX,XXX,XXX
```

Example - on Tuesday, Thursday and Friday this product can be purchases for all stores:

```
DAY = TUE,THU,FRI
```

—

Consecutive days use:

```
DAY = XXX-XXX
```

Example - from Tuesday to Thursday this product can be purchases for all stores:

```
DAY = TUE-THU
```

—

Expression use:

```
DAY = XXXXXX
```

Example - on weekends this product can be purchases for all stores:

```
DAY = WEEKENDS
```

Example - on weekday this product can be purchases for all stores:

```
DAY = WEEKDAYS
```

—

Mixed days and Expression use:

```
DAY = XXX,XXX-XXX,XXXXXX
```

Example - on Monday, Wednesday to Thursday, and weekends, this product can be purchases for all stores:

```
DAY = MON,WED-THU,WEEKDAYS
```

—

All above support the ~ for NOT. eg:

Example - on Tuesday this product can NOT be purchases for all stores:

```
DAY = ~TUE
```

Example - on weekends this product can NOT be purchases for all stores:

```
<code>DAY = ~WEEKENDS
```

MONTH RESTRICTIONS

Restricts the purchase of the product to the specified day(s) of the month.

Single day of month use:

```
MONTH = XM
```

Example - on the 16th day of the month this product can be purchases for all stores:

```
MONTH = 16
```

—

Multiple days of month use:

```
MONTH = XM,XM,XM
```

Example - on the 10th, 16th and 18th days of the month this product can be purchases for all stores:

```
MONTH = 10,16,18
```

—

Consecutive days of month use:

```
MONTH = XM-XM
```

Example - from the 20th to the 24th day of the month this product can be purchases for all stores:

```
MONTH = 20-24
```

—

Mixed days of month use:

```
MONTH = 10,16,18,20-24
```

Example - on the 10th, 16th and 18th days and of the month, and from the 20th to the 24th day of the month, this product can be purchases for all stores:

```
MONTH = 10,16,18,20-24
```

—

All above support the ~ for NOT. eg:

Example - on the 16th day of the month this product can NOT be purchases for all stores:

```
MONTH = ~16
```

Example - from the 20th to the 24th day of the month this product can NOT be purchases for all stores:

```
MONTH = ~20-24
```

DATE RESTRICTIONS

Restricts the purchase of the product to the specified date(s).

Single date use:

```
DATE = DDMONYYYY
```

Example - on the 01-JAN-2012 this product can be purchases for all stores:

```
DATE = 01JAN2012
```

—

Consecutive date use:

```
DAY = DDMONYYYY-DDMONYYYY
```

Example - from the 1st of January 2012 to the 31st of January 2012 this product can be purchases for all stores:

```
DAY = 01JAN2012-31JAN2012
```

—

Before date use:

```
DATE = BEFORE DDMONYYYY
```

Example - before the 31st of January 2012 this product can be purchases for all stores:

```
DATE = BEFORE 31JAN2012
```

—

After date use:

```
DATE = AFTER DDMONYYYY
```

Example - after the 31st of January 2012 this product can be purchases for all stores:

```
DATE = AFTER 31JAN2012
```

—

Single date supports the ~ for NOT. eg:

Example - on the 16th day of the month this product can NOT be purchases for all stores:

```
DATE = 31JAN2012
```

TIME RESTRICTIONS

Restricts the purchase of the product to the specified time(s).

Default time use:

```
TIME = HH:MM-HH:MM
```

Example - from 10am to 2:30pm this product can be purchases for all stores:

```
TIME = 10:00-14:30
```

—

Trade periods use:

```
TIME = TRADEPERIOD
```

Example - during breakfast trade period this product can be purchases for all stores:

```
TIME = BREAKFAST
```

Example - during lunch trade period this product can be purchases for all stores:

```
TIME = LUNCH
```

Example - during dinner trade period this product can be purchases for all stores:

```
TIME = DINNER
```

—

Before time use:

```
TIME = BEFORE HH:MM
```

Example - before 2:30pm this product can be purchases for all stores:

```
TIME = BEFORE 14:30
```

—

After time use:

```
TIME = AFTER HH:MM
```

Example - after 2:30pm this product can be purchases for all stores:

```
TIME = AFTER 14:30
```

—

Not before time use:

```
TIME = NOTBEFORE HH:MM
```

Example - before 2:30pm this product can NOT be purchases for all stores:

```
TIME = NOTBEFORE 14:30
```

—

Not after time use:

```
TIME = NOTAFTER HH:MM
```

Example - after 2:30pm this product can NOT be purchases for all stores:

```
TIME = NOTAFTER 14:30
```

—

Trade periods support the ~ for NOT. eg:

Example - during breakfast trade period this product can NOT be purchases for all stores:

```
TIME = ~BREAKFAST
```

TOPPING LIST SCRIPTS

Provides toppings and product options during product customisation.

IMPORTANT

Any toppings list scripts used must be defined under rules (excluding title scripts). eg:

```
ONLINETOPPINGRULE = TOPPINGS,SINGLECHOICE,SINGLECHOICE2,EXTRAS
```

—

TOPPINGS

Add a toppings list to a product (products selected by default). eg:

```
TOPPINGS = P,P,P,P
```

Add a custom title for the toppings list. eg:

```
TOPPINGLISTTITLE = Toppings list title
```

—

TOPPINGLISTSINGLECHOICE#

Add a maximum of 3 single choice toppings dropdowns to a product (single product selection is required). eg:

```
TOPPINGLISTSINGLECHOICE = P,D,T
TOPPINGLISTSINGLECHOICE2 = P,D,T
TOPPINGLISTSINGLECHOICE3 = P,D,T
```

Add a custom title for the single choice toppings dropdowns. eg:

```
TOPPINGLISTSINGLECHOICETITLE = First single choice dropdown title
TOPPINGLISTSINGLECHOICE2TITLE = Second single choice dropdown title
TOPPINGLISTSINGLECHOICE3TITLE = Third single choice dropdown title
```

—

TOPPINGLISTADDITIONAL

Add additional toppings/products dropdown to a product (product selections are optional). eg:

```
TOPPINGLISTADDITIONAL = P,D
```

Add a custom title for the additional toppings/products dropdown. eg:

```
TOPPINGLISTADDITIONALTITLE = Extra toppings dropdown title
```

Set the default product for the additional toppings/products dropdown. eg:

```
TOPPINGLISTADDITIONALDEFAULTOPTION = P
```

—

TOPPINGLISTEXTRAS

Add extra toppings/products dropdown to a product (product selections are optional). eg:

```
TOPPINGLISTEXTRAS = P,D
```

Add a custom title for the extra toppings/products dropdown. eg:

```
TOPPINGLISTEXTRASTITLE = Extra toppings dropdown title
```

Set the default product for the extra toppings/products dropdown. eg:

```
TOPPINGLISTEXTRASDEFAULTOPTION = P
```

—

FREETOPPINGS

Add free toppings/products dropdown to a product (product selections are optional). eg:

```
FREETOPPINGS = P,D
```

Add a custom title for the free toppings/products dropdowns. eg:

```
FREETOPPINGSLISTTITLE = Free toppings dropdown title
```

OLODEAL

Defines deals that are only available online.

Default use:

```
OLODEAL = P, P
```

Example - deal includes

- any one product in the same department as P1001
- any one product in the same department as P1002
- can be purchased online only
- final price is the the deal's product price
- available for all stores:

```
OLODEAL = P1001,P1002
```

—

Advanced use:

```
OLODEAL = P, T, D, CE
```

Example - deal includes

- any one product product in the same department as P1001
- any one matrix product with a target product T1357
- any one product from department D246
- any one product defined as combo element 125DRINK
- can be purchased online only
- final price is the deal product price
- available for all stores:

```
OLODEAL = P1001, T1357, D246, 125DRINK
```

—

Online deals support prefixes used in POS. eg: ' ! FORCE PRODUCT, can be used !Pxxx ' * WATERMARK, the price of this element is not changed to zero, but marked as part of the combo ' (%) WATERMARK, the price of this element is a percentage of the normal price, but marked as part of the combo ' ($) WATERMARK, the price of this element is a set price, but marked as part of the combo

Example - deal includes:

- ONLY the product P1001
- can be purchased online only
- final price is the deal's product price
- available for all stores:

```
OLODEAL = !P1001
```

Example - deal includes

- any one product in the same department as the watermarked product P1001 at the price of the product selected
- ONLY the product P1002
- can be purchased online only

- final price is deal's product price + the price of the watermarked product selected
- available for all stores:

```
OLODEAL = *P1001,!P1002
```

Example - deal includes

- any one product in the same department as the watermarked product P1001 at 75% of the price of the product selected
- can be purchased online only
- final price is 75% of the price of the watermarked product selected
- available for all stores:

```
OLODEAL = (%75)P1001
```

Example - deal includes

- any one product in the same department as the watermarked product P1001 for $10
- can be purchased online only
- final price is the watermarked price $10
- available for all stores:

```
OLODEAL = ($10)P1001
```

EXPIRES

Sets an expiry date for the product. The product cannot be purchased AFTER this date.

Note:
Available in cloud POS only atm.

Default use:

```
EXPIRES = DD-MON-YYYY
```

Example - product expires and cannot be used after 1 January 2012 for all stores:

```
EXPIRES = 01-JAN-2012
```

Struct use:

```
EXPIRES [ABC] = DD-MON-YYYY
```

Example - product expires and cannot be used after 1 January 2012 for stores using price struct SPECIALPRICES:

```
EXPIRES [SPECIALPRICES] = 01JAN2012
```

From:
<https://wiki.imagatec.com.au/> - **Imagatec Wiki**

Permanent link:
https://wiki.imagatec.com.au/doku.php?id=customer_portal:manuals:iorder:onlineproductscripts

Last update: **2025/07/14 14:48**