



# dPally URCap

---

User Manual

Version 3.1.0

# Table of Contents

<b>1 - License</b>	<b>3</b>
<b>2 - Introduction</b>	<b>4</b>
2.1 - License	4
2.2 - Installation	4
2.3 - Calibration	4
2.3.1 - Sensor placement and logic	4
2.3.2 - Calibration points	5
2.3.3 - Convert Pally program to dPally program	5
2.4 - Functional differences	5
2.5 - Callbacks	6
2.6 - Interpretation of pattern data	6
2.6.1 - Multi-pick and the Pattern verification state	6
2.6.2 - Locked gripper rotation	6
2.7 - Other limitations	6
2.8 - Scripts to convert Pally program to dPally program	7

# 1 - License

© Rocketfarm AS 2022. All rights reserved.

## Disclaimer

The information contained herein is the property of Rocketfarm AS and shall not be reproduced in whole or in part without prior written approval of Rocketfarm AS. The information herein is subject to change without notice and should not be construed as a commitment by Rocketfarm AS. Rocketfarm AS assumes no responsibility for any errors or omissions in this document.

The dPally logo is a registered trademark of Rocketfarm AS.

## 2 - Introduction

The dPally URCap is a depalletizer version of the Pally URCap. For a more detailed description, see the Pally URCap User Manual.

For easy identification of the supported features and known bugs, Pally and dPally share the same version information and release notes. Please check the Pally release notes for the latest new features, bug fixes, and compatibility issues.

This document describes only the differences between Pally and dPally.

### 2.1 - License

The dPally URCap requires its own license. The file name has the following format:  
`no.rocketfarm.urcap.depalletizer.RobotSerialNo.license`

Test licenses have the following name:

`no.rocketfarm.urcap.depalletizer.license`

### 2.2 - Installation

Both Pally and dPally can be installed on the same robot simultaneously. In this case two separate license files are required.

When installing Pally and dPally on the same robot, make sure the Daemon Port Number settings for the pallet manager daemon are different. The daemon processes will fail to start if the port numbers are identical.

It is technically possible to install different versions of Pally and dPally on the same robot, however, some unknown compatibility issues may occur.

### 2.3 - Calibration

The calibration process is similar to Pally. Conveyors are called "Drop-off" positions in dPally.

#### 2.3.1 - Sensor placement and logic

When using sensors on the conveyor, inverse logic might be required. This means the program can proceed with dropping when there are **no** other boxes on the conveyor. Make sure to select the 'inverse' checkbox next to the IO signals where needed.

With respect to the reversed direction of the conveyor, compared to Pally, the product sensors need to be mounted at the opposite end of the boxes. A signal should indicate that the box has left the drop-off position completely and there is room for the next box(es).

### 2.3.2 - Calibration points

When calibrating the drop position, make sure that the boxes will not collide with the conveyor. Leave a small gap between the gripper foam and the box surface during calibration.

The dPally URCap has a built-in force-controlled motion when picking boxes from the pallet. The robot moves above the box pick point and then vertically down until a specific force is detected. It is therefore recommended to calibrate the pallet positions with respect to this extra motion, i.e. leave a small gap between the gripper foam and the box surface during calibration.

### 2.3.3 - Convert Pally program to dPally program

Converting an existing Pally calibration to dPally is easy; just follow the steps below:

- make a copy of the existing URP program that was made with Pally
- create a new dPally program node in the program
- move the robot to each Pally calibration point by pressing 'Move here', then
- select the corresponding calibration point in dPally and press 'Set point'
- make sure to leave a small gap between the gripper foam and the box surface for each calibration point as described earlier
- remove the old Pally program node from the program when finished.

## 2.4 - Functional differences

The dPally URCap can depalletize from two different pallet positions (left and right). The following use cases are supported:

- depalletizing one pallet at a time, with a predefined pattern, on one conveyor
- depalletizing one pallet at a time, with a predefined pattern, on two conveyors
- depalletizing two pallets simultaneously on two conveyors (Dual Product mode)

The operator has to confirm when a **full pallet** is available on the left or right position. The program starts depalletizing the specified pallet only after operator confirmation.

The following table shows the main differences in the Pally-dPally terminology:

Pally	dPally
Palletizing	Depalletizing
Product arrival	Product leaving
Pickup position	Drop-off position
Above-pickup position	Vertical exit position
Empty pallet confirmation	Full pallet confirmation
Full pallet warning	Empty pallet warning

## 2.5 - Callbacks

Callbacks are fully supported by dPally. When using a custom path, however, there is a slight difference to be considered.

- `onNextTask`: called before the calculations of the next box start. Set `MovePerformed=True` to be able to use custom movement towards the pick position on the pallet.
- `beforeGrab`: called when the robot is about to pick a box from the pallet. Use custom motion to the pick position if needed.
- `afterGrab`: called when the robot is holding a box. Use custom motion to the drop position (conveyor) if needed.
- `afterRelease`: called when the robot has dropped the box on the conveyor. Use custom motion back to the waiting position if needed.

## 2.6 - Interpretation of pattern data

Pally and dPally can work with the same patterns. It should be possible to load and use the same pattern in Pally and dPally without problems, but there might be some differences.

### 2.6.1 - Multi-pick and the Pattern verification state

When a pattern has multi-pick and single-pick positions in one layer, dPally may choose a different box combination for the multi-picks. Example:

- A layer contains 3 boxes in a row
- Pally multi-picks boxes 1+2 and finally single-picks box 3
- dPally multi-picks boxes 3+2 and finally single-picks box 1

As shown above, palletizing a full pallet with Pally would never single-pick box position 1. When Pally and dPally are installed on the same robot, the patterns should be tested and verified with both.

### 2.6.2 - Locked gripper rotation

When creating a pattern for Pally in Pallet Builder, it is intuitive how to choose "enforced gripper rotation" at the pick position. However, when using the same pattern in dPally, the enforced gripper rotation should be interpreted as the gripper rotation at the **drop** position. Keep in mind the parameter defines the gripper rotation on the **conveyor** regardless of picking from- or dropping on it.

## 2.7 - Other limitations

Currently there are no other known limitations.

## 2.8 - Scripts to convert Pally program to dPally program

This is for developers only.

### Sample script to convert URP:

```
cat 7632.dual.UR.urp | gunzip | sed
's/installation="/installation="dPally./g' | sed
's/installationRelativePath="/installationRelativePath="dPally./
g' | sed 's/"Pally"/"dPally"/g' | sed
's/\.palletmanager\.Palletizer/\.depalletizer\.Depalletizer/g' |
sed
's/\.palletmanager\.UserCallback/\.depalletizer\.UserCallback/g' |
gzip > dPally.7632.dual.UR.urp
```

### Sample script to convert installation:

```
cat default.dual.installation | gunzip | sed
's/fileName="/fileName="dPally./g' | sed
's/"Pally"/"dPally"/g' | sed 's/palletmanager/depalletizer/g'
| sed 's/PalletManager/Depalletizer/g' | gzip >
dPally.default.dual.installation
```

Please note: replace the file names in the scripts to your own file names accordingly.