Upon receipt of the product and prior to initial operation, read these instructions thoroughly, and retain for future reference.
MANDATORY

- This manual explains the error recovery function. Read this manual carefully and be sure to understand its contents before operation.
- General items related to safety are listed in Section 1: Safety, in the NX100 Instructions. To ensure correct and safe operation, carefully read the NX100 Instructions before reading this manual.
- For detailed instructions regarding additional equipment including HyperStart, NX100 controller, manipulator, or other components, refer to the specific equipment manuals included with your documentation package.

CAUTION

- Some drawings in this manual are shown with the protective covers or shields removed for clarity. Be sure all covers and shields are replaced before operating this product.
- The drawings and photos in this manual are representative examples and differences may exist between them and the delivered product.
- YASKAWA may modify this model without notice when necessary due to product improvements, modifications, or changes in specifications. If such modification is made, the manual number will also be revised.
- If your copy of the manual is damaged or lost, contact a YASKAWA representative to order a new copy. The representatives are listed on the back cover. Be sure to tell the representative the manual number listed on the front cover.
- YASKAWA is not responsible for incidents arising from unauthorized modification of its products. Unauthorized modification voids your product's warranty.
- Software described in this manual is supplied against licensee only, with permission to use or copy under the conditions stated in the license. No part of this manual may be copied or reproduced in any form without written consent of YASKAWA.
Notes for Safe Operation

Before using this product, read this manual and all the other related documents carefully to ensure knowledge about the product and safety, including all the cautions.

In this manual, the Notes for Safe Operation are classified as "WARNING", "CAUTION", "MANDATORY", or "PROHIBITED".

⚠️ WARNING
Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury to personnel.

⚠️ CAUTION
Indicates a potentially hazardous situation which, if not avoided, could result in minor or moderate injury to personnel and damage to equipment. It may also be used to alert against unsafe practices.

⚠️ MANDATORY
Always be sure to follow explicitly the items listed under this heading.

🚫 PROHIBITED
Must never be performed.

Even items described as "CAUTION" may result in a serious accident in some situations.

At any rate, be sure to follow these important items

To ensure safe and efficient operation at all times, be sure to follow all instructions, even if not designated as "CAUTION" and "WARNING".
WARNING

• Before operating the manipulator, check that servo power is turned OFF pressing the emergency stop buttons on the front door of the NX100 and the programming pendant. When the servo power is turned OFF, the SERVO ON LED on the programming pendant is turned OFF. Injury or damage to machinery may result if the emergency stop circuit cannot stop the manipulator during an emergency. The manipulator should not be used if the emergency stop buttons do not function.

Fig. : Emergency Stop Button

• Once the emergency stop button is released, clear the cell of all items which could interfere with the operation of the manipulator. Then turn the servo power ON. Injury may result from unintentional or unexpected manipulator motion.

Fig. : Release of Emergency Stop

• Observe the following precautions when performing teaching operations within the P-point maximum envelope of the manipulator:
  – View the manipulator from the front whenever possible.
  – Always follow the predetermined operating procedure.
  – Ensure that you have a safe place to retreat in case of emergency. Improper or unintended manipulator operation may result in injury.

• Confirm that no person is present in the P-point maximum envelope of the manipulator and that you are in a safe location before:
  – Turning on the power for the NX100.
  – Moving the manipulator with the programming pendant.
  – Running the system in the check mode.
  – Performing automatic operations. Injury may result if anyone enters the P-point maximum envelope of the manipulator during operation. Always press an emergency stop button immediately if there is a problem.

The emergency stop buttons are located on the right of front door of the NX100 and the programming pendant.
Caution

- Perform the following inspection procedures prior to conducting manipulator teaching. If problems are found, repair them immediately, and be sure that all other necessary processing has been performed.
  - Check for problems in manipulator movement.
  - Check for damage to insulation and sheathing of external wires.
- Always return the programming pendant to the hook on the cabinet of the NX100 after use.
  The programming pendant can be damaged if it is left in the manipulator's work area, on the floor, or near fixtures.
- Read and understand the Explanation of Warning Labels in the NX100 Instructions before operating the manipulator:
Notation for Menus and Buttons

Descriptions of the programming pendant, buttons, and displays are shown as follows:

<table>
<thead>
<tr>
<th>Item</th>
<th>Manual Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu</td>
<td>The menus displayed on screen are denoted with { }, ex. {TOOL}.</td>
</tr>
<tr>
<td>Button</td>
<td>The buttons, check boxes, radio buttons displayed on screen are denoted with [], ex. [Close]; [Sync] check box; [Fast] radio button.</td>
</tr>
</tbody>
</table>

Description of the Operation Procedure

In the explanation of the operation procedure, the expression "Select • • • " means the following operations:

- To move the cursor to the object item and left-click on it with the mouse.
- To pick out the object item by the tab key and press the Enter key.
  (In case of selecting a menu, use arrow keys instead of the tab key to pick out the object item, then press the Enter key.)

Registered Trademark

In this manual, names of companies, corporations, or products are trademarks, registered trademarks, or brand names for each company or corporation. The indications of (R) and TM are omitted.
1 Introduction .....................................................................................................................................1-1
   1.1 Overview................................................................................................................................1-1
   1.2 About this Document...............................................................................................................1-3
   1.3 Hardware Layout....................................................................................................................1-4
      1.3.1 System Requirements ....................................................................................................1-5
      1.3.2 EasyPallet® Hardware Components........................................................................1-5
      1.3.3 EasyPallet® Software Components........................................................................1-5
      1.3.4 Optional Software .........................................................................................................1-5
   1.4 Learning EasyPallet® ..........................................................................................................1-5
      1.4.1 XML Resources ...........................................................................................................1-5
   1.5 Reference to Other Documentation .....................................................................................1-6

2 Setup and Configuration .............................................................................................................2-1
   2.1 Defining Your Palletizing Cell ............................................................................................2-1
   2.2 Connecting Monitor, Keyboard, and Mouse ......................................................................2-2
   2.3 Tool Setup ..........................................................................................................................2-2
      2.3.1 Mount Gripper ............................................................................................................2-2
      2.3.2 Define Tool Data .......................................................................................................2-3
      2.3.3 Gripper I/O Job Setup ..............................................................................................2-6
   2.4 Creating a New Gripper Configuration File .......................................................................2-7
   2.5 User Frame File Setup .......................................................................................................2-7
      2.5.1 Modified TCP for Teaching User Frames ................................................................2-8
      2.5.2 Conveyor User Frame Requirements: ......................................................................2-9
      2.5.3 Pallet User Frame Requirements: ............................................................................2-10
      2.5.4 Slipsheet Dispenser User Frame Requirements: ..................................................2-11
   2.6 Defining T-Axis Orientation ..............................................................................................2-13
   2.7 Transferring Robot Files to Embedded PC .......................................................................2-15
      2.7.1 Saving UserFrame Files .............................................................................................2-16
      2.7.2 Saving Parameter Files ...............................................................................................2-17
      2.7.3 Transferring Files to Embedded PC .........................................................................2-18
   2.8 Create Pallet Jobs and Check for T-Axis Limits ..................................................................2-19
   2.9 Fixed Vacuum Gripper ......................................................................................................2-21
      2.9.1 Creating a New Fixed Suction Gripper Configuration File ....................................2-22
      2.9.2 Modifying the Fixed Suction Gripper Project File ..................................................2-23
2.9.3 Modifying Gripper Configuration Files at Build Time .................................................. 2-23
2.9.4 Switching between Fixed Suction and Standard Suction Cup Gripper .................. 2-23
2.10 Fixed Fork Gripper ........................................................................................................ 2-24
  2.10.1 Gripper X-Axis Offset ........................................................................................ 2-25
  2.10.2 Gripper Y-Axis Offset ........................................................................................ 2-26
  2.10.3 Gripper Z-Axis Offset ........................................................................................ 2-26
  2.10.4 Setting Up the Fixed Fork Gripper TCP ........................................................... 2-27
  2.10.5 Creating a Fixed Fork Gripper Configuration File ........................................... 2-28
  2.10.6 Modifying the FixedFork Gripper Project File ................................................... 2-29
  2.10.7 Changing Gripper Configuration Files at Build Time ........................................ 2-29
2.11 Multiple Cell Support ..................................................................................................... 2-29
2.12 Project File and Editing Summary ................................................................................. 2-31
  2.12.1 Editing the I/O Groups ...................................................................................... 2-34

3 EasyPallet® Pendant Operation .............................................................................................. 3-1
  3.1 Starting EasyPallet® (Pendant) .................................................................................... 3-1
  3.2 Understanding the EasyPallet® Pendant Interface ....................................................... 3-2
    3.2.1 Main Screen ......................................................................................................... 3-2
      3.2.1.1 File Menu ................................................................................................ 3-2
      3.2.1.2 Project Configuration Area ...................................................................... 3-3
      3.2.1.3 Build Miscellaneous Options ................................................................... 3-4
      3.2.1.4 Pallet Configuration ................................................................................. 3-4
      3.2.1.5 Create Jobs ............................................................................................. 3-7
      3.2.1.6 Minimize App ........................................................................................... 3-7
      3.2.1.7 Close App ................................................................................................ 3-7
      3.2.1.8 Status ...................................................................................................... 3-7
    3.2.2 Box Editor ............................................................................................................. 3-8
    3.2.3 Cell Setup User Interface ..................................................................................... 3-8
      3.2.3.1 Conveyors ............................................................................................... 3-9
      3.2.3.2 Pallets ..................................................................................................... 3-9
      3.2.3.3 SlipSheets ............................................................................................. 3-10

4 EasyPallet® Control Panel (PC Based) .................................................................................. 4-1
  4.1 Main Control Panel ........................................................................................................ 4-2
    4.1.1 File Menu ............................................................................................................ 4-2
    4.1.2 [Re-Load All Databases] ..................................................................................... 4-2
    4.1.3 Project File Edit and Select Functions ............................................................... 4-2
    4.1.4 Build Major Options ........................................................................................... 4-2
    4.1.5 Build Misc. Options ............................................................................................ 4-4
4.1.5.1 Keep Order ..............................................................................................4-4
4.1.5.2 Look Ahead Jobs ....................................................................................4-4

4.1.6 View Commands .....................................................................................4-5

4.2 Pallet Pattern Editor ....................................................................................4-6
  4.2.1 Edit Functions .........................................................................................4-6
  4.2.2 Pallet Table with Preview .......................................................................4-6

4.3 Layer Pattern Editor ....................................................................................4-7
  4.3.1 Layer Pattern - Edit and Select Functions ............................................4-7
  4.3.2 Layer Pattern Elements ..........................................................................4-7
  4.3.3 Box Order Over Ride ..............................................................................4-8
  4.3.4 Layer Pattern Table with Preview .........................................................4-11

4.4 Box Editor (PC Based) ................................................................................4-12
  4.4.1 XML Editing Tools ................................................................................4-12
  4.4.2 Display Tools .........................................................................................4-12
  4.4.3 Box Definition Table ..............................................................................4-12

4.5 Empty Pallet Editor .....................................................................................4-13
  4.5.1 Edit Tools ..............................................................................................4-13
  4.5.2 Display Tools .........................................................................................4-13
  4.5.3 Empty Pallet Definition Table ...............................................................4-13

4.6 Batch Build .................................................................................................4-14
  4.6.1 Batch Build Edit and Save Functions ....................................................4-14
  4.6.2 Batch Build Control Panel ..................................................................4-14

4.7 Manual Sequencer ......................................................................................4-15
  4.7.1 Layer Manual Sequence File -- Edit and Select Functions ....................4-15
  4.7.2 Layer Info Tab .......................................................................................4-15
  4.7.3 Pick and Place Editor ............................................................................4-16
  4.7.4 Pick Tab ...............................................................................................4-20
  4.7.5 Place Tab ..............................................................................................4-24

5 Operation ......................................................................................................5-1
  5.1 Overview ..................................................................................................5-1
  5.2 Process Flow .............................................................................................5-2
    5.2.1 File Synchronization ...........................................................................5-2
  5.3 Embedded PC Directory Structure ..........................................................5-3
    5.3.1 Database and Application Directories .................................................5-3
5.3.2 Embedded PC - Cell and Job Archive Directories ................................................ 5-6
5.3.3 Embedded PC - Robot Subdirectory .................................................................... 5-6
5.3.4 Embedded PC - Job and Command Archive Files ............................................... 5-7
5.4 Pendant CF Card Directory Structure ........................................................................ 5-8
5.5 EasyPallet® Conversion Overview ................................................................................ 5-9
  5.5.1 Master Job Structure .......................................................................................... 5-10
5.6 Robot on a Rail Applications ........................................................................................ 5-10
  5.6.1 Robot on a Rail Setup - Static Rail Option ......................................................... 5-11
  5.6.2 Robot on a Rail Setup - Dynamic Rail Option .................................................... 5-12
5.7 Creating New Jobs Using the Pendant Application ......................................................... 5-13
5.8 Modifying Jobs Using the Pendant Application ............................................................ 5-15
5.9 Easy Pallet Pick and Place Sequence Options ............................................................... 5-15
6 Advanced Programming (XML) ................................................................................................. 6-1
  6.1 Job Templates ........................................................................................................... 6-1
    6.1.1 Job Numbering Scheme ..................................................................................... 6-5
    6.1.2 XML Template to Robot Job Conversion Process .............................................. 6-6
    6.1.3 Meta Commands and Variables ........................................................................... 6-6
      6.1.3.1 Meta Command and Variable Examples ....................................................... 6-7
      6.1.3.2 Processing Instructions ............................................................................... 6-7
    6.1.4 Editing a Template File ...................................................................................... 6-8
      6.1.4.1 Example 1: Modifying the Pickup Sequence ............................................... 6-8
      6.1.4.2 Example 2: Adding a Call to Load Empty Pallet ........................................ 6-11
    6.1.5 MasterTemp.XML ............................................................................................. 6-12
    6.1.6 Velocity Control with D Variables ..................................................................... 6-12
  6.2 Box Definition Database ............................................................................................... 6-13
    6.2.1 BoxDB.XML .................................................................................................. 6-14
  6.3 Gripper Definitions Database ....................................................................................... 6-14
    6.3.1 GripperDB.XML ............................................................................................. 6-14
  6.4 Pallet Definition Database ........................................................................................... 6-27
    6.4.1 PalletDB.XML .................................................................................................. 6-27
1 Introduction

1.1 Overview

EasyPallet® is part of the Motoman family of standardized solutions. It is a fully integrated hardware/software system, supported by Motoman, Inc. The EasyPallet® system features an embedded PC that enables creation and modification of palletizing jobs directly from the programming pendant in less than 5 minutes with as little as 3 menu or button selections.

Using predefined XML files that define the robot cell, including standard pallets and layer patterns, EasyPallet® is able to provide rapid creation and modification of palletizing jobs directly from the programming pendant. EasyPallet® is able to create full featured jobs that include automatic vectoring, automatic asynchronous placement, and user selectable job formats. Job approach and departure motion types and velocity are also easily controlled by EasyPallet®. EasyPallet® pendant operation is easy to perform with little robotic knowledge required.

Advanced Users

EasyPallet® offers unparalleled flexibility, with a hybrid programming model that offers many additional capabilities for the Advanced User. These additional functions are accessed from the EasyPallet® Control Panel that runs on the embedded PC.

From the control panel, the advanced user can directly modify the EasyPallet® project files that enable EasyPallet® to support up to 8 pallet stations, 8 in-feed conveyor stations and a robot in a fixed location or on a rail.

The advanced user can also create additional generic layer patterns and change the type or form of a palletizing job by modifying existing job template files or creating new ones.

Batch processing provides the ability to create an unlimited set of palletizing jobs from the EasyPallet® control panel. This can be crucial for rebuilding jobs after a major robot crash or during cell modification.

The creation of generic Manual Sequence files provides complete control over the pick and place sequence. This provides a number of benefits including the ability to slightly increase the operational envelope of the cell, as well as minimizing the cycle time by rotating the gripper at high speeds when the robot is returning to the pickup conveyor.

The following table provides a summary of the types of functionality available to the advanced user.

<table>
<thead>
<tr>
<th>Function Sub-Function</th>
<th>Basic Operator</th>
<th>Integrator/Advanced Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job Creation Create Palletizing Jobs</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Change Job structure (Job Templates)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Limited Box Order Control</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Maintain Box Orientation</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Job Template Creation/Modification</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Advanced users can add new pallet definitions, box definitions, and pallet patterns by modifying the XML files directly from the embedded PC.

### NOTE

Modifications to the XML files must comply with the XML schema as defined in Section 6 "Advanced Programming (XML)".

It is recommended that any changes to the XML files be made to copies of the files and not to the originals.

The EasyPallet® interface runs directly on the programming pendant, and accesses powerful palletizing software running on the embedded PC via an Ethernet link with FTP file transfer protocol. This is an open standard used for file transfer between computers that are connected on a LAN (Local Area Network), or the Internet.

*Fig. 1-1: EasyPallet® Pendant Interface*
1.2 About this Document

This manual is intended as an introduction and overview for personnel who are familiar with the operation of their Motoman robot model and Microsoft® Windows®/PC usage.

This User’s Manual provides an overview of the Motoman EasyPallet® system. For detailed information on specific system components listed in this document, please refer to the documentation package included with your EasyPallet® system.

This manual contains the following sections:

Section 1 - Introduction
This chapter introduces the EasyPallet® User’s Manual, provides an overview of the EasyPallet® system, lists reference documents that are included with the documentation package, and provides Motoman Customer Support contact information.

Table 1-1: EasyPallet® Features

<table>
<thead>
<tr>
<th>Function</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robots</td>
<td>Supports 4, 5, and 6 axis robots</td>
</tr>
<tr>
<td>Robot on a Rail</td>
<td>Supports upright or inverted robot mounting</td>
</tr>
<tr>
<td>Robot Controller</td>
<td>NX100 only</td>
</tr>
<tr>
<td>Grippers</td>
<td>• Vacuum grippers up to 4 Boxes, Side to Side or End to End&lt;br&gt;• Clamp&lt;br&gt;• Shovel&lt;br&gt;• Clam Shell</td>
</tr>
<tr>
<td>Conveyors</td>
<td>Supports up to 8 conveyors</td>
</tr>
<tr>
<td>Layer Patterns</td>
<td>• Independent of box and pallet definitions&lt;br&gt;• Over 200 pre-defined patterns included&lt;br&gt;• Based on palletizing industry standards&lt;br&gt;• Pre-viewable from programming pendant&lt;br&gt;• Ability to rotate to form interlocking pallet patterns&lt;br&gt;• No reprogramming required to change box or pallet sizes</td>
</tr>
<tr>
<td>Job Format</td>
<td>• Joint&lt;br&gt;• Relative Job&lt;br&gt;• Standard Master Job Set&lt;br&gt;• Plug in pattern jobs&lt;br&gt;• Ability to store multiple job patterns and call from one Master job</td>
</tr>
<tr>
<td>Job Features</td>
<td>Automatic Vectoring and Asynchronous Pick and Place&lt;br&gt;• Eliminates box clipping when building tight patterns&lt;br&gt;• Saves operator build time&lt;br&gt;• Saves robot palletizing build time</td>
</tr>
<tr>
<td>Job Customization</td>
<td>XML Job templates provide extensive job customization.</td>
</tr>
<tr>
<td>Slipsheets</td>
<td>Supports up to 4 slipsheet stations</td>
</tr>
<tr>
<td>NX100 Controller</td>
<td>API Enabled version required. Contact Motoman for latest software version supported.</td>
</tr>
<tr>
<td>Software Version</td>
<td>64 Meg Pendant Recommended</td>
</tr>
</tbody>
</table>
1 Introduction
1.3 Hardware Layout

Section 2 - Setup and Configuration
This section provides setup and configuration instructions for the EasyPallet® system.

Section 3 - EasyPallet® Pendant Operation
This section provides basic operating procedures for the EasyPallet® pendant application.

Section 4 - EasyPallet® Control Panel (PC Based)
This section provides basic operating procedures for the control panel application running on the embedded pc.

Section 5 - Operation
This section provides instructions EasyPallet® operation including creating, modifying, and deleting pallet jobs directly from the programming pendant.

Section 6 - Advanced Programming (XML)
This section provides job templates and instructions for modifying template files.

1.3 Hardware Layout

An embedded industrial PC and Ethernet switch are mounted directly in the robot controller. The computer uses powerful palletizing simulation software along with XML schemas to rapidly create pallet patterns.

The EasyPallet® software runs directly on the programming pendant accessing the simulation software via FTP file transfer protocol. Once system setup files have been defined (including; pallet, conveyor, gripper orientation, userframe files, and gripper I/O) palletizing jobs can be created and modified directly from the programming pendant.

Pallet configuration and gripper tool data is sent directly from the programming pendant to the embedded PC via the Ethernet switch using FTP file transfer protocol. Using fully modifiable XML schemas and powerful simulation software, the data is manipulated and robot *.JBI job files are created. Once the new *.JBI job files have been created, they are available for use on the NX100 controller directly from the programming pendant.

Fig. 1-2: Hardware Layout
1.3.1 System Requirements

- A 64MEG Programming Pendant
- Controller software - NS3.71(JP_US)-44
- Compact Flash* memory card (ATA Flash)
  - "CFI-***MBA" made by HAGIWARA sys-com.
  - "SDCFBI-**.****" made by SUN DISK.
- Compact Flash* Card Reader with USB 2.0 support
- USB 2.0 keyboard
- Monitor - supported by MS-Windows (256 colors or more) with a minimum resolution of 1200 X 800
- USB 2.0 mouse

* Compact Flash is a registered trademark of SUN DISK.

1.3.2 EasyPallet® Hardware Components

- Programming Pendant
- Pendant Ethernet Cable
- Embedded PC
- Ethernet Switch
- Power Supply
- Ethernet Connections

1.3.3 EasyPallet® Software Components

- EasyPallet® Pendant software
- Microsoft XML NotePad
- EasyPallet® Control Panel
- Filezilla FTP Server

1.3.4 Optional Software

The MotoSim software packages is available for use with the EasyPallet®:

1.4 Learning EasyPallet®

Motoman provides a variety of options to help you to learn to use EasyPallet®, including training and technical support. For more information on available training classes for EasyPallet®, please contact our training department at: training@motoman.com or visit our website at: www.motoman.com.

1.4.1 XML Resources

There are so many resources related to XML that we can't possibly list them all here. However, here is a short list of books and online resources you may find useful.

**XML Books**
- XML: A Primer - by Simon St. Laurent, MISPress, 1998
- XML Applications - by Frank Boumphrey et.al., Wrox Press, 1998

**XML Online Resources**
- Microsoft's XML Center (msdn.microsoft.com/en-us/xml/default.aspx)
- XML.org - (http://www.xml.org)
1.5 Reference to Other Documentation

For additional information refer to the following:

- Motoman MotoSim EG Manual (P/N 152002-1)
- Motoman NX100 Controller Manual (P/N 149201-1)
- Motoman NX100 Maintenance Manual (P/N 150133-1)
- Motoman NX100 Operator’s Manual for General (P/N 150077-1)
- Motoman NX100 Operator’s Manual for Handling (P/N 149231-1)
- Motoman NX100 Concurrent I/O Parameter Manual (P/N 149230-1)
- Motoman NX100 Independent/Coordinated Control Function Manual (P/N 149648-1)
- Motoman INFORM User’s Manual (P/N 150078-1)
- Vendor manuals for system components not manufactured by Motoman
2 Setup and Configuration

While the physical installation of the integrated pc, and installation of the software are performed by a Motoman technician, there are other procedures that you must complete before EasyPallet® can be used with your palletizing system. During installation, your Motoman technician will setup the software for your particular robot including simulation files defining your robot. A robot folder is automatically created within the EasyPallet® directory so that the simulation software can access defining files for your palletizing cell.

The robot folder is installed by default in `C:\Program Files\Motoman\EPMotsim\RobotModel`. Robotmodel is your robot model type. Typical models include:

- **EPL160-A00**
- **HP165-A00**

**CAUTION**

The customer is responsible for providing trained operators to run the equipment. The customer is also responsible for making sure that all equipment is operated in accordance with the ANSI/RIA R15.06-1999 Robot Safety standard, as well as any other local or state standards.

2.1 Defining Your Palletizing Cell

After the Motoman technician has completed the installation process, you must define your palletizing cell. This process includes:

- 2.3 "Tool Setup" - Mount gripper and calibrate robot TCP.
- 2.3.3 "Gripper I/O Job Setup"
- 2.5 "User Frame File Setup" - Create User Frames for the pallet, conveyor, and gripper.
- 2.6 "Defining T-Axis Orientation" - Create orientation jobs the pallet and conveyor.
- 2.7 "Transferring Robot Files to Embedded PC".
- 2.8 "Create Pallet Jobs and Check for T-Axis Limits".

**CAUTION**

The palletizing cell setup must be performed in the order shown above. It is very important that setup is completed in this order as each operation builds upon the previous. For example, you must have a well defined TCP before accurate user frames can be defined. Also, orientation jobs require both a gripper TCP and user frames before they can be taught.
2.2 Connecting Monitor, Keyboard, and Mouse

To define your palletizing cell, it is necessary to connect a keyboard, monitor, and mouse to the embedded PC.

WARNING

Before connecting, be sure to turn the main power supply OFF, and put up a warning sign (ex. DO NOT TURN POWER ON). Be sure to follow all Lock Out/Tag Out procedures and comply with the National Electrical Code and/or local electrical codes.

Failure to observe this warning may result in electric shock or injury.

To connect your keyboard, monitor, and mouse, proceed as follows:

1. Open the front door of the robot controller.
   a) Insert key in the door lock on the front of controller (two places), and turn 90 degrees clockwise.
   b) Rotate the main power supply switch to the "OFF" position and open the door.

2. Confirm that the primary power supply is OFF.

3. Remove spare access plate from side of controller and run monitor, keyboard, and mouse cables into the controller.

4. Re-attach the plate and cables firmly so that it won’t shift or slide out of place.

5. Route the cables inside the controller to the embedded pc.

6. Connect monitor, keyboard, and mouse cables to the appropriate connections on the embedded pc (refer to vendor documentation for connection specifics).

7. Close the controller door.
   a) Close the door gently.
   b) Rotate the door locks counterclockwise 90 degrees.

2.3 Tool Setup

This section describes how to mount and setup your gripper for EasyPallet® operation. It is very important to ensure that the gripper tool data correctly reflects the characteristics of the gripper being used.

2.3.1 Mount Gripper

To mount and setup your gripper, proceed as follows:

1. Move the robot to Home position. Refer to Section 8.1 of your NX100 Controller manual for detailed information (P/N 149201-1).

   The Robot T-Axis should be at approximately 0 pulse counts.
2. Mount the gripper to the robot flange plate according to the manufacturer’s mounting instructions and recommendations. Generally, the gripper is mounted with the gripper long axis parallel to either the robot X- or Y-axis.

2.3.2 Define Tool Data

Once the gripper has been mounted, the tool center point (TCP), weight, center of gravity and other tool data must be defined. Tool files are used to record this gripper data. The TCP data includes the difference between the default location of the robot TCP at the robot T-axis flange and the new TCP location for the gripper. The typical vacuum gripper TCP is defined at the top center point of the group of boxes that is being carried with the X-axis parallel with the longest axis of the gripper.

*Fig. 2-3: Typical Vacuum Gripper TCP*

*Fig. 2-4: Typical Shovel TCP*
The clam shell TCP is typically the same as the vacuum gripper except located at the bottom of the box.

While the gripper TCP location is well defined, it is not always easy to visualize. It is highly recommended that a threaded pointer be created and attached to the gripper to help define the TCP.

---

**WARNING**

The gripper pointer must be removed before normal operation. Failure to remove the pointer can cause a robot crash or damage to the palletizing cartons, or other system components.
Define the robot tool data according to the following requirements. Refer to Section “8.3 Tool Data Setting” in the NX100 Controller manual (P/N 149201-1) for detailed instructions on tool calibration TCP.

- TCP X and Y - Defined so the tool rotates about the gripper pointer (for most grippers this will be TCP X and Y of approximately 0.0, as most grippers are mounted at the center of the mounting flange of the robot.

- TCP Z - Defined at same height as the bottom of the working surface of the vacuum cups.

- TCP Rx - Set to 0.0 degrees.

- TCP Ry - Set to 0.0 degrees.

- TCP Rz - Defined so the TCP X-Axis is aligned with or parallel to the longest length of the vacuum gripper.

- Tool Load Data - Refer to Section “8.4.3 Tool Load Information Setting” in the NX100 Controller manual (P/N 149201-1).

To define the robot tool data, proceed as follows:

1. Working from the programming pendant, select [ROBOT] under the main menu.

2. Select [TOOL]. The TOOL COORDINATE window appears.

3. Select the desired tool number using the arrow key and press Select. The TOOL window appears.

4. Select the desired tool data to modify using the arrow key.

5. Enter tool data and press [Enter].

When setup properly, the gripper TCP should look similar to Fig. 2-8: “Gripper TCP”. Note that the robot TCP X-axis is aligned with the Gripper's longest dimension and that positive TCP Z is pointing down, away from the gripper.

Fig. 2-8: Gripper TCP
2 Setup and Configuration
2.3 Tool Setup

2.3.3 Gripper I/O Job Setup

Two gripper I/O jobs are used to map the B-variables used by EasyPallet® to the I/O bits used to turn the gripper suction cups ON or OFF. These B-variables trigger the appropriate I/O ON and OFF, activating the suction cups to lift the boxes, and deactivating to release. These I/O must be correctly mapped for your system.

EasyPallet® uses one B-variable per box to control pick up or release. Typically, a GripOn job is used to pick up boxes from the conveyor. The following four B-variables are typically used:

- B013 for Box 1
- B014 for Box 2
- B015 for Box 3
- B016 for Box 4

When the robot is in position to pick up a set of boxes from the conveyor, the GripOn job is called. The GripOn job must contain logic to check the state of each B-variable. With a standard gripper of 4 boxes or less, each B-variable is assigned a value of 1 or 0 to signify I/O state needed to actuate and release the gripper vacuum (one (1) being ‘Vacuum On’, and zero (0) representing the ‘Vacuum Off’ state). Box 1 suction cups are controlled by B011, Box 2 suction cups are controlled by B012 and so on.

The following example shows a typical GripOn job for a 4 box gripper. In this example, Box 1 suction cups are controlled by Output 20, Box 2 cups are controlled by output 21, and so on.

**GRIPON.JBI Example**

NOP
JUMP *skipBox1 IF B011<>1
DOUT OT#(20) ON
*skipBox1
JUMP *skipBox2 IF B012<>1
DOUT OT#(21) ON
*skipBox2
JUMP *skipBox3 IF B012<>1
DOUT OT#(22) ON
*skipBox3
JUMP *skipBox4 IF B013<>1
DOUT OT#(23) ON
*skipBox4

When the robot tool is in position to drop one or more boxes, the GripOff job is called and 1 to 4 boxes maybe dropped.

**GRIPOF.JBI Example**

NOP
JUMP *skipBox1 IF B011<>0
DOUT OT#(20) OFF
*skipBox1
JUMP *skipBox2 IF B012<>0
DOUT OT#(21) OFF
*skipBox2

Each system can be quite different. For example, depending on the gripper, one I/O bit may control a single suction cup or it may control 2 or even 4 suction cups.
2.4 Creating a New Gripper Configuration File

Gripper configuration files contain extra setup parameters for fixed clamper and vacuum style grippers. See chapter 2.9 and chapter 2.10 for more detailed information.

1. Go to the directory: C:\Program Files\Motoman\EasyPalletPC2\DB_Grippers\Dir_FixedForkorClamper GripperConfiguration
2. Copy and Paste one of the existing files.
3. Rename the new file.
4. Open the file with the XML Notepad Editor.
5. Change the values as required.
6. Save the changes you have made.
7. Shut down and restart the EasyPallet Server.
8. Shut down and restart the EasyPallet Control Panel.

The new Configuration File appears in the EasyPallet® Control Panel, Gripper Config. Combo Box.

2.5 User Frame File Setup

EasyPallet® uses User Frames to convert between the robot coordinate system and pallet and conveyor coordinate systems. Before you can create safe and accurate palletizing jobs that can be used in a real cell, you must make sure the user frames used in EasyPallet® are identical to those used in the actual cell. To do this, create a user frame file for every component of your cell (pallets, conveyors, slipsheet dispenser) using the programming pendant and import these files to the EasyPallet® folder on the embedded pc.
User frame coordinates are defined by three points that have been taught to the manipulator through axis operations. These three defining points are ORG, XX, and XY, as shown in the diagram below. These three points of positional data are registered in a user coordinate file (UFrame.cnd).

ORG is the home position. XX is a point on the X-axis. XY is a point on the Y-axis side of the user coordinates that has been taught, and the directions of Y- and Z-axes are determined by point XY.

It is important that the two points, ORG and XX, be taught accurately.

Refer to Section 8.8 “User Coordinate Setting” in the NX100 Controller Manual (P/N 149201-1) for detailed information regarding User Frames.

### 2.5.1 Modified TCP for Teaching User Frames

A userframe must be defined for every component of your cell (pallets, conveyors, slipsheet dispenser) and copied to the EasyPallet® folder on the embedded pc. Because most ORG positions cannot be reached using the defined TCP, a modified TCP can be used to program the user frames for the various system components. A threaded pointer can be created and attached to the gripper to help extend the TCP and the robot tool data modified to reflect this change.

1. Create a threaded pointer of a known length to extend the TCP.
2. From the programming pendant, select [ROBOT] under the main menu.
3. Select [TOOL]. The TOOL COORDINATE window appears.
4. Select the desired tool number using the arrow key and press Select. The TOOL window appears.
5. Modify the Z-axis tool data to match the extended TCP pointer and press [Enter].

**WARNING**

The extended TCP gripper pointer must be removed and robot tool data corrected before normal operation. Failure to remove the pointer and adjust robot tool data can cause a robot crash or damage to the palletizing cartons, or other system components.

### 2.5.2 Conveyor User Frame Requirements:

- **X-Axis** is parallel to the longest Conveyor Axis
- **Y-Axis** is parallel to the shortest Conveyor Axis
- User frame defined on top of the rollers or conveyor surface (i.e. where the boxes ride)
- User frame defined where the box stops, assuming the box is right justified on the conveyor. This location is at the intersection of the Right Hand Side Stop and the End Stop.

**Fig. 2-10: Conveyor User Frame**
2.5.3 Pallet User Frame Requirements:

- X-axis parallel to the long side of the pallet.
- Y-Axis parallel to the short side of the pallet.
- User frame defined at the bottom of the pallet.
- User frame defined in one of two corners with the X-axis of the user frame parallel to the pallet's longest axis.

Fig. 2-11: Pallet User Frame
2.5.4 Slipsheet Dispenser User Frame Requirements:

• The X-axis must be parallel to the long side of the slipsheet dispenser.

• The Y-axis must be parallel to the short side of the slipsheet dispenser.

• Place the user frame in one of two corners that allows the X-axis of the user frame to be parallel to the slipsheet dispenser’s longest axis.

To create a userframe uframe.cnd file, proceed as follows:

1. Select ROBOT under the main menu.

2. Select USER COORDINATE. The USER COORDINATE window appears.

3. Select the user coordinate number using the cursor and press SELECT. The setting window appears.

<table>
<thead>
<tr>
<th>DATA</th>
<th>EDIT</th>
<th>DISPLAY</th>
<th>UTILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER COORDINATE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USER COORD NO.: 01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SET POS: CRD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOOL: 00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L : &lt; STATUS &gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U : ORG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R : XX : O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B : XY : O</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. Select “SET POS”.

Fig. 2-12: Negative Y-Axis User Frame
5. Select the teaching point.

6. Move the manipulator to the desired position using the axis keys.

7. Press [MODIFY] then [ENTER]. The taught position is registered.

8. Repeat the steps 5 to 7 to teach the ORG, XX and XY positions.

9. “●” indicates that teaching is completed and “○” indicates that it is not completed.

10. Select “COMPLETE”. The User Coordinate window appears with the User Coordinate Set complete.

WARNING

You must remove the extended TCP gripper pointer and correct the robot tool data before continuing. Failure to remove the pointer and adjust robot tool data can cause a robot crash or damage to the palletizing cartons, or other system components.
2.6 Defining T-Axis Orientation

For most robot models, the robot T-Axis has over 360 degrees of movement. Therefore, many of the conveyor pickup and pallet setdown orientations (0, 90, 180, and 270 degrees) have multiple T-Axis values. Each of these orientations must be recorded for each pallet, conveyor, and slipsheet dispenser used in the cell. These orientations are recorded in xml files located in the dir_orient folder. EasyPallet® uses these orientations to calculate the best path between locations and minimize unnecessary rotation of the tool when moving between the conveyors, pallets, and slipsheet dispenser. Without these orientation files, the robot T-Axis may rotate the tool unnecessarily, causing potential cable damage, hose leakage, and dropped product.

To define your T-axis orientations, you must access the embedded PC operating system using a monitor, keyboard, and mouse. Make certain the monitor, keyboard and mouse are properly installed before proceeding with the following instructions (refer to chapter 2.2 "Connecting Monitor, Keyboard, and Mouse" at page 2-2).

To record your T-axis orientations, proceed as follows:
1. Move the manipulator to the desired position using the programming pendant.
2. Select ROBOT under the main menu.
3. Select CURRENT POSITION. The CURRENT POSITION window appears.
4. Select Pulse coordinate.
5. Rotate the T-axis in the negative direction until the pulse limit is reached.
6. With the T-axis at the negative pulse limit, rotate the T-axis in the positive direction to the first orientation available (0, 90, 180, or 270 degrees) and record the axis position data (SLURBT).
7. Continue rotating the T-axis in the positive direction, recording each orientation (0, 90, 180, and 270 degrees) until the positive soft limit is reached.
8. Repeat steps 1-4 recording the T-axis orientations for every pallet, conveyor, and slipsheet dispenser used in your cell.

Fig. 2-13: Pallet Orientation

Enter the position values recorded for every orientation in the appropriate .xml file (pallet1.xml, conveyor1.xml, etc.).
1. Locate and open the appropriate orientation file on the embedded pc. (c:\program files\motoman\motosimag\cells\epmotsim\dir_orient\pallet1.xml)

2. Modify the .xml file, modifying <Angle> tag and axis tags with previously recorded data.

```xml
<Root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="C:\Program Files\Motoman\EasyPalletPC2\Dir_AllXMLSchemas\Orient_JobsV1.xsd">
  <UserFrametoTCPXSet Active="False">
    <Angle>180</Angle>
    <S>107702</S>
    <L>92232</L>
    <U>-47087</U>
    <R>0</R>
    <B>-7248</B>
    <T>-107823</T>
  </UserFrametoTCPXSet>
  <UserFrametoTCPXSet Active="False">
    <Angle>270</Angle>
    <S>107702</S>
    <L>92232</L>
    <U>-47087</U>
    <R>0</R>
    <B>-7248</B>
    <T>-69823</T>
  </UserFrametoTCPXSet>
  <UserFrametoTCPXSet Active="False">
    <Angle>0</Angle>
    <S>107702</S>
    <L>92232</L>
    <U>-47087</U>
    <R>0</R>
    <B>-7248</B>
    <T>-32823</T>
  </UserFrametoTCPXSet>
  <UserFrametoTCPXSet Active="True">
    <Angle>180</Angle>
    <S>-37886</S>
    <L>120306</L>
    <U>-130587</U>
    <R>0</R>
    <B>0</B>
    <T>-145996</T>
  </UserFrametoTCPXSet>
  <UserFrametoTCPXSet Active="True">
    <Angle>270</Angle>
    <S>-37886</S>
    <L>120306</L>
    <U>-130587</U>
    <R>0</R>
    <B>0</B>
    <T>-65996</T>
  </UserFrametoTCPXSet>
</Root>
```

3. For Conveyors and slipsheet dispensers, select the 4 desired orientations to be used and change the UserFrametoTCPXSet tag to “True” and all other orientations to “False”.

```xml
<UserFrametoTCPXSet Active="True">
  <Angle>180</Angle>
  <S>-37886</S>
  <L>120306</L>
  <U>-130587</U>
  <R>0</R>
  <B>0</B>
  <T>-145996</T>
</UserFrametoTCPXSet>
<UserFrametoTCPXSet Active="True">
  <Angle>270</Angle>
  <S>-37886</S>
  <L>120306</L>
  <U>-130587</U>
  <R>0</R>
  <B>0</B>
  <T>-65996</T>
</UserFrametoTCPXSet>
```
2.7 Transferring Robot Files to Embedded PC

Once you have created orientation and userframe files for each component of your system (pallets, conveyors, slipsheet dispenser and gripper), as well as jobs defining gripper I/O, you must transfer the following files from the robot controller to the embedded PC:

- All.prm
- Tool.cnd
- UFrame.cnd

Insert a compact flash memory card (ATA Flash) in the compact flash slot on the programming pendant. To insert a compact flash into the programming pendant,

1. Open the cover on the rear of the programming pendant.
2. Insert the compact flash so that the product label is on the front side. The card cannot be inserted in any other position. Do not forcibly insert the card.
3. After inserting the card, be sure to close the cover before starting operation.

CAUTION

Observe the following precautions when using the compact flash:

- Handle the compact flash with care to protect the stored data.
- Do not use or keep the compact flash in places where strong static electricity or electronic noise may occur.
- Do not drop or exert any shock or strong force to the compact flash.
- Do not remove the compact flash or turn OFF the power when accessing the compact flash (writing-in or reading-out the compact flash data). The data in the compact flash may be lost.
- Back up the data from the compact flash to other media such as floppy disks or hard disks.
2.7.1 Saving UserFrame Files

To transfer the UserFrame condition files from the memory of the NX100 to the external memory device, perform the following procedure:

1. Select FD/CF CARD from the main menu.
2. Select SAVE. The following window appears.

3. Select FILE/GENERAL DATA. The selection window appears.

4. Select the USER COORDINATE DATA and TOOL DATA.cnd files to be saved. The selected files are marked with "★".

5. Press [ENTER]. The confirmation dialog box appears.
6. Select “YES.” Saving starts. To cancel saving, select “STOP.” Once saving is complete or cancelled, the selection window for condition files or general data appears.

2.7.2 Saving Parameter Files

To transfer the ALL.PRM batch parameter file from the memory of the NX100 to the external memory device, perform the following procedure:

1. Select FD/CF CARD from the main menu.
2. Select SAVE. The following window appears.
3. Select PARAMETER. The selection window for parameters appears.
4. Select the BATCH PARAMETER ALL.PRN file to be saved. Selected parameters are marked with a “*”.
2.7 Transferring Robot Files to Embedded PC

2.7.3 Transferring Files to Embedded PC

To transfer the pallet, conveyor, slipsheet dispenser, and gripper orientation and userframe files from the compact flash memory card to the embedded PC, you must access the embedded PC operating system using a monitor, keyboard, mouse, and compact flash to USB 2.0 card reader/adaptor. Make certain the monitor, keyboard and mouse are properly installed before proceeding with the following procedure (refer to chapter 2.2 "Connecting Monitor, Keyboard, and Mouse" at page 2-2).

1. Remove the compact flash memory card from the slot on the programming pendant.
2. Insert the compact flash memory card into the compact flash card reader with USB and plug the USB cable into any of the USB ports on the embedded PC.
3. Double click the My Computer icon on the embedded PC desktop and locate the compact flash memory card.

4. Transfer the following files from the compact flash directory to the robot folder.
   - UFRAME.CND
   - TOOL.CND
   - ALL.PRN

5. Press [ENTER]. The confirmation dialog box appears.

   ![Save confirmation dialog box]

6. Select “YES.” Saving starts. To cancel saving, select “STOP.” Once saving is complete or cancelled, the selection window for parameters appears.

**WARNING**

When accessing Motoman program files, do not move or rename these files. EasyPallet® will not function correctly without access to these folders.
2.8 Create Pallet Jobs and Check for T-Axis Limits

Once EasyPallet® has been setup to be used with your palletizing system, create a few sample jobs to test for T-Axis limit and other reach errors.

1. Start EasyPallet® from the programming pendant.

2. Using the Project dropdown menu in the Project Configuration area, select the job template you wish to use to create your robot jobs. Job templates determine coordinate system and job structure and can be modified to meet your specific needs. See Appendix A, for more detailed information.

3. Select the gripper being used from the Tool dropdown menu.

4. Begin building the pallet layers using the Pallet Configuration area. The Pallet Pattern dropdown menu can be used to load existing pallet patterns for use and modification.
   a) Starting with Layer 1 as the bottom layer, select the Box Name and Layer Pattern from the dropdown menus.
   b) Continue building pallet layers using the Add and Remove Layer buttons and the Box Name and Layer Pattern dropdown menus.

5. Select the pallet being used for this job using the Pallet dropdown menu.

6. Name your new pallet configuration by clicking the [Edit] button. The on screen keyboard appears.

7. Using the on screen keyboard, enter the name of the pallet configuration and press Enter.

8. Once you have defined your pallet configuration, click the [Create Jobs] button. The “Send Build Request - waiting for job files from server...” message appears in the Status window. Files are sent from

The robot folder is installed by default in c:\program files\motoman\epmotsim\robotmodel. Where robotmodel is your robot model type. Examples include: EPL160-A00, HP165-A00.
the programming pendant to the embedded pc and the simulation software begins processing the job request.

9. If the simulation software is able to create the jobs, the message “Last Action: saved requested files” appears in the status window and the set of .jbi robot job files are uploaded to the programming pendant and are available for use.

Errors can occur in the job build process for a variety of reasons. The most common error (Error #9 Build Error) occurs due to reach issues with the robot.

2.9 Fixed Vacuum Gripper

You can modify the tool pickup location from the center of the gripper or gripper TCP to a fixed location on the gripper by modifying the Fixed Vacuum Gripper Configuration files. Fixed locations are typically located at one of the corners of the gripper.
Once the tool pickup location has been modified, EasyPallet® examines the gripper pickup pattern (number of boxes and configuration) and automatically offsets the pickup location so the pickup group is attached to the gripper at the new fixed location.

EasyPallet® calculates this offset by adjusting the robot pick and place locations before the robot jobs are created. This avoids the use of P-variable offsets.

Modifying the tool pickup location allows you to accommodate grippers that have uneven suction due to gripper design (i.e. location and size of the gripper plenums). This can also be used to change the functional TCP of the palletizing system thereby modifying the effective reach of the palletizing system.
Before modifying the tool pickup location, verify correct gripper setup (see Section 2.3 "Tool Setup"). Additionally, the following step must be performed:

1. Create a fixed vacuum gripper configuration file.
2. Link the new gripper configuration file to the project file.
3. Create a pallet build with the new project file.

### 2.9.1 Creating a New Fixed Suction Gripper Configuration File

Fixed suction gripper configuration files are located in the DB_Gripper directory as shown below. In the example below, there are 4 configuration files; each setting the reference corner at a different corner of the gripper. However there is no limit to the number of configuration files you can create.

**Fig. 2-15: Fixed Suction Gripper Configuration Files**

Each configuration file must use one of the generic GripperTypeName elements listed in the GripperDB.xml file. In the example above, the "VacFixed1BoxEtoE" element is used.

Gripper length and width are defined as well as the fixed reference point locations for the X and Y-axes. In the example, the Fixed reference point is set to X = 310 and Y = 200. This is the positive X and the positive Y corner of the gripper.

### 2.9.2 Modifying the Fixed Suction Gripper Project File

Once the new fixed gripper configuration file has been created, it must be added to the project file. To do so, an optional "ToolConfiguration" element must be added after the "ToolName" element in the xml file.
2.9 Fixed Vacuum Gripper

2.9.3 Modifying Gripper Configuration Files at Build Time

You can change the gripper configuration file directly from the Build Major Options window of the EasyPallet® Control Panel by selecting the desired file from the Gripper Config. dropdown window (highlighted in yellow below).

![Fixed Suction Gripper](image)

The following example is the same build only with a normal suction or vacuum gripper selected. Note that the Gripper Config. selection is disabled.

2.9.4 Switching between Fixed Suction and Standard Suction Cup Gripper

It is often useful to switch between a Fixed Suction Gripper and Standard Suction Type gripper between builds. This can be done directly from the EasyPallet Control panel on a build-by-build basis by switching between a Suction Cup and a Fixed Suction Cup Gripper in the Select Tool dropdown menu in the Build Major Options window (highlighted in yellow below).

Fig. 2-17: Fixed Suction Gripper

The following example is the same build only with a normal suction or vacuum gripper selected. Note that the Gripper Config. selection is disabled.
2.10 Fixed Fork Gripper

Fixed Fork or Clamper grippers have one fixed Fork and one or more movable forks. These types of grippers require EasyPallet® to calculate an offset from the normal TCP X-, Y-, and sometimes Z-axes to properly pickup and place boxes.

With this type of gripper, there are always two sides of the gripper that may collide with boxes already placed on the pallet. In general, the offsets are calculated to have the boxes aligned with two sides (the control sides) of the gripper.

*Fig. 2-18: Fixed Fork Gripper*

When using Fixed Fork type grippers, select the following features and options:

- Box Order Over Ride
- Keep Box Orientation,
- Keep Order Build
2.10 Fixed Fork Gripper

2.10.1 Gripper X-Axis Offset

The gripper X-axis offset is set in one of two ways:

- For single place sequences, the gripper picks up the boxes with the negative X gripper edge aligned with the first box on the conveyor.
- For asynchronous drops, the boxes are centered along the gripper X-axis.

The table and figures below explain the reasons for using this strategy. The software automatically creates these X Offsets.

<table>
<thead>
<tr>
<th>Pickup - For Single Drop</th>
<th>Pickup for Asynchronous Drop</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Boxes are aligned with end of X-Axis</td>
<td>- Boxes are Centered along X-Axis</td>
</tr>
<tr>
<td>- Allows the gripper to approach boxes previously placed on the pallet from 2 directions</td>
<td>- Aligns the boxes with the moveable gripper fork</td>
</tr>
<tr>
<td></td>
<td>- Asynchronous drops must be the first boxes placed on the layer</td>
</tr>
</tbody>
</table>
2.10.2 Gripper Y-Axis Offset

Most fixed fork grippers are designed so that the fixed fork position can be mechanically moved to accommodate smaller or larger sets of boxes.

EasyPallet® allows you to set the fixed fork distance directly in the configuration file. The following figures show how the fixed distance is defined. Once a configuration file is setup and selected in the project file, EasyPallet® calculates the gripper Y offset required to align the box edge with the gripper’s fixed fork surface.

Fig. 2-21: Fixed Fork Gripper

2.10.3 Gripper Z-Axis Offset

EasyPallet® also determines if a Z-axis offset is required. Z-axis offsets are required for boxes that are shorter than the “Finger” plus a safety clearance distance of 1 inch.

Fig. 2-22: Fixed Fork Gripper with Short Box And Z-Axis Offset
The following steps must be completed to setup and use a Fixed Fork Gripper.

1. Setup the tool TCP.
2. Create a fixed fork gripper configuration file.
3. Link to the gripper configuration file in the project file.
4. Create a pallet build using the project file.

### 2.10.4 Setting Up the Fixed Fork Gripper TCP

1. Locate the gripper TCP at the center of the robot T-axis and on a plane located where the top of the boxes are held on the gripper (see Figure below). Typical TCP coordinates: X = 0, Y = 0, Z = 150 to 300 mm.

2. The TCP Tz value must be adjusted so that the X-Axis is parallel to the long axis of the gripper (see Figure below).

*Fig. 2-23: Fixed Fork Gripper TCP*
2.10.5 Creating a Fixed Fork Gripper Configuration File

The Fixed Fork Gripper Configuration Files are located in the DB_Gripper directory. There is no limit to the number of configuration files you can create. See the figures below.

Fig. 2-24: Fixed Fork Configuration Files

The following figure shows a fixed fork gripper configuration file. The most important variables are highlighted in yellow.

Fig. 2-25: Fixed Fork Gripper Configuration File

```
<Root>
  <GripperTypeName>SymmetricalForclBoxStcGripperTypeName</GripperTypeName>
  <GripperTCPNumber>0</GripperTCPNumber>
  <GripperNegLength>457.2</GripperNegLength>
  <GripperPosLength>457.2</GripperPosLength>
  <GripperFingerWidth>300</GripperFingerWidth>
  <GripperFingerHeight_MountingFlangeToFingerTop>364.92</GripperFingerHeight_MountingFlangeToFingerTop>
  <GripperFingerHeight>311.49</GripperFingerHeight>
  <Distance_GripperFingerToFixedFinger_Y_Minimum>316.48</Distance_GripperFingerToFixedFinger_Y_Minimum>
  <Distance_GripperFingerToFixedFinger_Y_CurrentSetup>316.48</Distance_GripperFingerToFixedFinger_Y_CurrentSetup>
  <MovedFingerData>
    <MovedFinger_Y_Minimum>323.00</MovedFinger_Y_Minimum>
    <Distance_MovedFinger_Y_CurrentSetup>323.00</Distance_MovedFinger_Y_CurrentSetup>
  </MovedFingerData>
  <Distance_BoxToFinger_GripperOpen>25.40</Distance_BoxToFinger_GripperOpen>
  <GripperFingerData>
  </GripperFingerData>
</Root>
```

The GripperFingerHeight element determines when the automatic Z-axis offset is required.

The Distance_GripperTCP_ToFixedFingerY_CurrentSetup element calculates the current gripper Y-axis offset for each box being palletized.
2.10.6 Modifying the FixedFork Gripper Project File

Once the new fixed fork gripper configuration file has been created, it must be added to the project file. To do so, an optional "ToolConfiguration" element must be added after the "ToolName" element in the xml file.

The ToolConfiguration element is only valid with Suction Fixed or Fixed Fork type grippers. To see a list of all gripper types, open the Generic Gripper file:
C:\Program Files\Motoman\EasyPalletPC2\DB_Grippers

2.10.7 Changing Gripper Configuration Files at Build Time

It is often useful to change the active gripper configuration file between builds. This can be done directly from the EasyPallet Control panel on a build-by-build basis by selecting the desired file from the Gripper Config. dropdown menu in the Build Major Options window (highlighted in yellow below).

2.11 Multiple Cell Support

A database management tool is available for those integrators or power users who need to support multiple robotic palletizing cells. Both the EasyPallet Control Panel and the EasyPallet PC Server software have the ability to select a database root directory.

To setup multiple palletizing cells:
   For Example: C:\Program Files\Motoman\EasyPalletPC2\FKI2
2. Copy all the DB Files from the EasyPallet root directory to the new directory.

3. Modify the new EasyPallet database files as required (some of the more standard files are listed below):
   - MotoSimEG Cell File -- C:\Program Files\Motoman\EasyPalletPC2\DB_MotoSimCell\MotoSimCellDB.xml - Replace the name of the new MotoSimEG simulation cell and robot type with new values.

   WARNING

Do NOT copy the following EasyPallet® EXE sub-directories:
C:\ProgramFiles\Motoman\EasyPalletPC2\EXE_EasyPalletPCControl Panel
C:\ProgramFiles\Motoman\EasyPalletPC2\EXE_EasyPalletServer2
2 Setup and Configuration

2.11 Multiple Cell Support

- EasyPallet Project File -- C:\Program Files\Motoman\EasyPalletPC2_FK12\Dir_EasyPalletProjects
  Create a new project File for your palletizing cell.

- Box Database -- C:\Program Files\Motoman\EasyPalletPC2_FK12\DB_Box
  Define boxes used with this cell

- Layer Patterns -- C:\Program Files\Motoman\EasyPalletPC2_FK12\Dir_LayerPatterns
  Create new layer Patterns as required.

- Pallet Patterns -- C:\Program Files\Motoman\EasyPalletPC2_FK12\Dir_PalletPatterns
  Create new pallet Patterns as required

4. Start EasyPallet PC Server and select the new directory

Fig. 2-26: EasyPallet® PC Server

5. Start EasyPallet® Control Panel and select the new directory.
The EasyPallet project file is a primary file that must be set up before palletizing jobs can be created. This file is the fundamental driver that fills the EasyPallet Pendant and the EasyPallet Control Panel with many values. It is so important that the vast majority of features and functions in the Pendant and the Control Panel are disabled until a project file is selected.

The EasyPallet project file holds two basic types of information: Cell Setup and Job Build Information. The cell Setup information provides a description of the number and type of conveyor, pallet, and slipsheet stations that are in the cell. It also provides detailed settings for each station such as the User Frame Number. See the Cell Setup User Interface topic in Section 3 for a detailed description of these setup parameters.

The Job Build Information is a collection of all the parameters that are needed to create a build. They are described in detail in both Sections 3 and 4.

Editing the Project File:

Parts or all of the project file can be edited with one of the following tools:

- Any third party XML Editor
- EasyPallet Pendant Application
- EasyPallet Control Panel

The following table and figures provide a summary of the Project File Features and function. The table also lists the application that can be used to edit this function. The figures have numbers and arrows that identify the location of each function that can be edited with that application.

<table>
<thead>
<tr>
<th>#</th>
<th>Parameter</th>
<th>Definition</th>
<th>Edit on Pendant</th>
<th>Edit on Embedded PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ToolName</td>
<td>Defines tool type to build pallet</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>LastPalletStationNumber</td>
<td>Defines Pallet Station to build the jobs. Valid entries are 1 to 8.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>LastPatternName</td>
<td>This is the last pallet pattern that was used to create the jobs. This file describes the pallet layer and box configuration.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>LastSequence</td>
<td>Not used - Deprecated</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>4</td>
<td>LastTemplate</td>
<td>Determines the form of the job by selecting a different set of templates to create the job.</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>LastJobName</td>
<td>Determines the archive job directory name on the Embedded PC</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>#</td>
<td>Parameter</td>
<td>Definition</td>
<td>Edit on Pendant</td>
<td>Edit on Embedded PC</td>
</tr>
<tr>
<td>----</td>
<td>---------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>6</td>
<td>LastJobBuild_PatternNumber</td>
<td>Defines the pattern number to place on the pallet jobs that are being created. This number will appear in the last 3 digits of the job name. Example: XYZ00AAA.jbi Where AAA is the pattern number.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>LastJobBuild_KeepBoxOrientation</td>
<td>Keeps box on pallet orientations 0, 180, 90 and 270 separate. Used for labels out.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>8</td>
<td>LastJobBuild_CreateLookAheadJobs</td>
<td>Creates jobs that tell the conveyor how many boxes to release for the next pickup</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>LastJobBuild_CreateMasterJob</td>
<td>Determines if the Master job and its helper jobs will be moved to the pendant with the next build request.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>10</td>
<td>LastJobBuild_BuildOrdered</td>
<td>Uses the Auto Sequence Planner but tries to maintain the Box Order Override</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>11</td>
<td>Conveyor Stations</td>
<td>Add/subtract Conveyor Stations (Cell Setup User Interface topic in Section 3)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>12</td>
<td>Pallet Stations</td>
<td>Add/SubTract Pallet Stations (Cell Setup User Interface topic in Section 3)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>13</td>
<td>Slip Sheet Stations</td>
<td>Add/Subtract SlipSheet Stations (Cell Setup User Interface topic in Section 3)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>14</td>
<td>Conveyor Select IO Group</td>
<td>From PLC to Robot Controller Determines the IO Group and size that EasyPallet will read to get the Pick From Conveyor Number</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>15</td>
<td>SlipSheet On Pallet Select IO Group</td>
<td>From PLC to Robot Controller Determines the IO Group and size that EasyPallet will read to get the SlipSheet on Pallet Station command</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
### EasyPallet®

#### Setup and Configuration

##### 2.12 Project File and Editing Summary

<table>
<thead>
<tr>
<th>#</th>
<th>Parameter</th>
<th>Definition</th>
<th>Edit on Pendant</th>
<th>Edit on Embedded PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>SlipSheet On Layer Select IO Group</td>
<td>Frm the PLC to Robot Controller Determines the IO Group and size that EasyPallet will read to get the Place Slipsheet on Layer command</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>17</td>
<td>Pallet Select IO Group</td>
<td>From the PLC - to Robot Controller Determines the IO Group and size that EasyPallet will read to get the Place on Pallet Station Number</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>18</td>
<td>Pallet LookAhead IO Group</td>
<td>Optional From Robot to PLC - Tells the PLC how many boxes to release on Conveyor X for the next robot pickup for Pallet Y</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>19</td>
<td>Pattern IO Group</td>
<td>From PLC to Robot Controller Determines which Pallet pattern (Goes into the last 3 digits of the Plugin Jobs)</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

*Fig. 2-27: Editing Project File with EasyPallet Control Panel*
2.12.1 Editing the I/O Groups

As you can see from the chart above, the I/O Groups are the only section that must be edited in a standard XML Editor. These values are generally a one time setup function and they are almost self explanatory.

The following excerpt shows the Conveyor Select IO Group. Generally the user will change the I/O Size from Full Group (8 bits) to Half Group (4 bits) and the IO Group Number to the desired Robot IO Group. The other variables should not be changed.

```
<MasterJob_ConveyorSelectIOGroup>
  <IOSize>FullGroup</IOSize>
  <IOGroupNumber>3</IOGroupNumber>
  <PUTInVariable>
    <VariableType>B</VariableType>
    <VariableIndex>3</VariableIndex>
  </PUTInVariable>
  <IOGroupName>ConveyorSelectIOGroup</IOGroupName>
  <IOType>Input</IOType>
</MasterJob_ConveyorSelectIOGroup>
```
3 EasyPallet® Pendant Operation

3.1 Starting EasyPallet® (Pendant)

This chapter provides basic operating instructions for the EasyPallet® pendant application. EasyPallet is an incredibly flexible and versatile tool. There are many ways to accomplish the same task depending on your preferences. Pallet jobs can be created, modified, and deleted working solely from the programming pendant, or the same work can be accomplished working directly with the embedded pc via a monitor, keyboard, and mouse. Some situations may warrant creating the pallet, box, and pattern definitions directly from the embedded pc using the EasyPallet® Control Panel. It may be easier for you to modify definitions directly in the xml files while creating the actual jobs using the programming pendant.

If you will take the time to read the following sections and become familiar with the various ways of interfacing with EasyPallet®, you will find your experience with EasyPallet® much more rewarding.

3.1 Starting EasyPallet® (Pendant)

The EasyPallet® pendant application is accessed from the programming pendant. To start EasyPallet®, proceed as follows:

1. From the Main menu on the programming pendant, click on the [PP Application] button.

![Programming Pendant Menu]

2. Select EasyPallet from the dropdown menu.
3.2 Understanding the EasyPallet® Pendant Interface

EasyPallet® features an easy to use graphic interface. The various screens, fields and buttons are described below.

3.2.1 Main Screen

3.2.1.1 File Menu

(Send Cell Files to Server)

The {Send Cell Files to Server} command manually uploads all local files from the programming pendant to the embedded pc. This is typically not necessary as EasyPallet® automatically sends required files during the Create Jobs process. However, if changes have been made to the programming pendant or compact flash memory card, it may be necessary to use the {Send Cell Files to Server} command to ensure that changes made at the programming pendant are reflected on the embedded pc.

(Refresh All Local Files)

The {Refresh All Local Files} command updates files on the programming pendant. If files have been changed on the embedded pc (i.e. database files, etc.) it is important to update or refresh those files on the programming pendant. The speed of the transfer depends on the quality of the compact flash memory device used and the type of file being transferred.

*NOTE*

The *.bmp files used to depict generic pallet patterns (C:\ProgramFiles\Motoman\EasyPalletPC\Dir_LayerPatterns) are approximately 400 times larger than the typical XML layer pattern file. Transferring one *.bmp file can take up to 30 to 60 seconds. Additionally, there are approximately 60 layer patterns that ship with EasyPallet®. Transferring all layer *.bmp files can take up to 30 minutes or more to complete.
3 EasyPallet® Pendant Operation
3.2 Understanding the EasyPallet® Pendant Interface

During the synchronization process, the Status window at the bottom of
the interface indicates: “Requested all files. (This operation may take
several minutes to complete...).” When all files have been synchronized
between the pc and programming pendant, the Status window indicates:
“Last Action: Saved requested files”.

3.2.1.2 Project Configuration Area

The Project Configuration area defines the project structure EasyPallet®
uses to create the robot jobs.

![Project Configuration Area](image)

{Project}

The {Project} dropdown menu selects the job template used to create the
robot jobs. Two sample job templates are included with EasyPallet®.

- Example2_Joint_Project template - creates jobs using the Joint
coordinate system. Each robot position is recorded using the pulse
counts for each manipulator axis (S,L,U,R,B,T).
- Example2_XYZ_Project template - creates relative jobs based on
the XYZ (Cartesian) coordinate system. Each robot position is
defined by the robot position and orientation from the origin of the
Cartesian coordinate system (X, Y, Z, Xz, Yz, Zz).

Each project template creates a Master job along with Pick, Place, Init,
and Inc robot jobs as described below:

- **Master job** - calls the Pick, Place, Init, and Inc jobs as required.
- **Pick job** - defines how product is picked up from the conveyor.
- **Place job** - defines how product is placed on the pallet.
- **Init job** - initializes counters at the start of the process.
- **Inc job** - increments counters and is updated after every pick and
place.

{Pallet #}

The {Pallet #} dropdown menu defines the pallet station number in a multi-
pallet cell.

{Job Pattern #}

The {Pallet #} dropdown menu defines the pallet pattern to be used on this
pallet station.

The creation of robot job files is controlled by a set of xml
template files. Advanced users familiar with XML
programming can modify these files and create new project
templates. See chapter 6 "Advanced Programming (XML)"
for more detailed information.
3 EasyPallet® Pendant Operation
3.2 Understanding the EasyPallet® Pendant Interface

{Tool}
The {Tool} dropdown menu allows the user to select existing gripper definitions to be used in the project. Gripper files are typically defined by the number of boxes and box alignment (end to end or side to side) as follows:

- Vac2BoxEtoE - 2 box gripper with 8 vacuum cups. Boxes are aligned end to end on the gripper.
- Vac4BoxStoS - 4 box gripper with 16 vacuum cups. Boxes are aligned side by side on the gripper.

Gripper definitions are stored in the GripperDB.xml file. Grippers can be added, modified and updated directly in this file. Refer to chapter 6.3 "Gripper Definitions Database" for additional information on adding or modifying gripper definitions.

[Save]
The [Save] button saves changes to the current project job template.

[Save As]
The [Save As] button allows the user to save changes to the current project under a new name using the on screen, virtual keyboard.

[Cell Setup]
The [Cell Setup] button opens the cell setup interface screen allowing the user to define their cell including pallet pattern, layer pattern, box definitions, empty pallet definitions, and cell setup (see chapter 3.2.3 "Cell Setup User Interface").

3.2.1.3 Build Miscellaneous Options

The Build Misc. Options frame provides options that help determine what jobs are included in the project.

{Create Master}
The {Create Master} selection allows the user to elect not to update and create the Master job, updating and creating only the pick, place, init, and inc. jbi job files. This can be very useful when making minor changes to the cell without shutting down production.

{Keep Build Order}
The {Keep Build Order} selection uses the EasyPallet Auto Sequence planner and Box Order Override sequence to plan the pick and place order for this pallet build.

{Keep Box Orient.}
The {Keep Box Orientation} selection maintains the default box orientation.

{Create Look Ahead Jobs}
The {Create Look Ahead Jobs} selection allows the user to create look ahead jobs with the PLC.
3.2.1.4 Pallet Configuration

The Pallet Configuration area allows the user to modify the pallet configuration of existing projects or enter new pallet configurations for entirely new pallet projects. The Pallet Configuration area is laid out with two columns and three rows depicting the pallet layers and the box names and layer patterns for each layer. The numbered dropdown menu is used to determine which middle layer is displayed when defining more than three pallet layers.

Box Name

The *Box Name* dropdown menu allows the user to select a particular box for each pallet layer. New box definitions can be input using the Box Editor (see chapter 3.2.2 "Box Editor").

Layer Pattern

The *Layer Pattern* dropdown menu allows the user to select pallet patterns for each layer. Over 150 pallet patterns have been pre-defined. However, new patterns can be added to EasyPallet® by advanced users familiar with xml programming (see Appendix A: Advanced Programming).

Rotate

The *Rotate* dropdown menu allows the user to quickly rotate the pallet patterns (0, 90, 180, and 270 degrees) for each layer.

Fig. 3-29: Pattern Rotate

Slip-sheet

The *Slip-sheet* radio button allows the user to quickly define a layer as a slipsheet.
The question mark button next to each layer pattern opens the pattern viewer screen. Pre-programmed box patterns can be viewed by selecting the pattern from the dropdown menu and clicking the [Display] button. Use the [Return] button to return to the main EasyPallet® screen.

(Pallet)
The {Pallet} dropdown menu allows the user to select a particular pallet to be used for the project.

(Pallet Pattern)
The {Pallet Pattern} dropdown menu allows the user to select from existing pallet patterns.

[Edit Button]
The [Edit] button allows the user to modify the Pallet Pattern name or enter a new pallet pattern name using the on screen, virtual keyboard.

[Add Layer]
The [Add Layer] button allows the user to add new layers to the top of the current pallet project.

[Remove Layer]
The [Remove Layer] button allows the user to remove the top layer of the current pallet project.

[Box Editor]
The [Box Editor] button opens the Box Editor screen allowing the user to define and name new box definitions using the on screen, virtual keyboard (see chapter 3.2.2 "Box Editor").

3.2.1.5 Create Jobs
The [Create Jobs] button sends the defining files to the PC simulation engine to create robot .JBI job files based on the input parameters.

3.2.1.6 Minimize App
The [Minimize App] button collapses the EasyPallet® screen to allow the user to access the main screen of the programming pendant. The EasyPallet® application remains running in the background.

3.2.1.7 Close App
The [Close App] button shuts down the EasyPallet® application.

3.2.1.8 Status
The Status window displays status messages during EasyPallet® operation.
3.2.2 **Box Editor**

The Box Editor allows the user to modify an existing box definition or enter new box configurations for entirely new boxes.

*Fig. 3-30: Box Editor*

- **{Box Name}**
  - The {Box Name} dropdown menu allows the user to select a predefined box to be edited.

- **[KeyPad]**
  - The [KeyPad] button allows the user to modify the box name or enter a new box name using the on-screen virtual keyboard.

- **{Length}**
  - The {Length} window allows the user to modify the box length.

- **{Width}**
  - The {Width} window allows the user to modify the box width.

- **{Height}**
  - The {Height} window allows the user to modify the box height.

- **{Save}**
  - The [Save] button saves changes made in the Box Editor directly to the BoxDB.xml database.

- **{Return}**
  - The [Return] button closes the Box Editor and returns the user to the main EasyPallet® screen.

3.2.3 **Cell Setup User Interface**

The cell setup user interface allows the user to define and configure the components of their cell. Conveyors, pallets, and slipsheet dispensers can be added or removed from the cell definition. Simple modifications can be made to the basic component definition including; userframe, Y-axis alignment, and box rotation.
3.2.3.1 Conveyors

The Conveyors frame allows the user to define the conveyors used in the cell (up to 8) including user frame, Y-axis alignment, and box rotation.

- Select the conveyor name using the dropdown menu (conveyor1, conveyor2, etc).
- Select the userframe file used for each conveyor using the dropdown menu. The conveyor userframe file is created during system setup. Refer to chapter 2.5.2 "Conveyor User Frame Requirements:" for additional information.
- Define the conveyor userframe Y-axis alignment as positive or negative using the dropdown menu. Y-axis alignment allows you to set the userframe with the negative Y-axis along the short conveyor axis.
- Set the box orientation on the conveyor using the dropdown menu (0,90,180,270).
- Additional conveyors can be added using the [+] button or deleted from the cell using the [-] button.

3.2.3.2 Pallets

The Pallets frame allows the user to define the pallet stations used in the cell (up to 8) including user frame, Y-axis alignment, conveyor feeds, and slip feeds.
3 EasyPallet® Pendant Operation

3.2 Understanding the EasyPallet® Pendant Interface

• Select the pallet station name using the dropdown menu.
• Select the userframe file used for each pallet station using the dropdown menu. The pallet userframe file is created during system setup. Refer to chapter 2.5.3 "Pallet User Frame Requirements:" for additional information.
• Define the pallet userframe Y-axis alignment as positive or negative using the dropdown menu. Y-axis alignment allows you to set the userframe with the negative Y-axis along the short conveyor axis.
• Select which conveyor feeds the pallet using the Conveyor Feeds dropdown menu.
• Select which slipsheet station feeds the pallet using the Slip Feeds dropdown menu.
• Additional pallet stations can be added using the [+] button, or deleted from the cell using the [-] button.

3.2.3.3 SlipSheets

The SlipSheets frame allows the user to define the slipsheets used in the cell (up to 4) including user frame, Y-axis alignment, and rotation.

• Select the slipsheet station name using the dropdown menu (slipsheet1, slipsheet2, etc).
• Select the userframe file used for each slipsheet station using the dropdown menu. The slipsheet userframe file is created during system setup. Refer to chapter 2.5.4 "Slipsheet Dispenser User Frame Requirements:" for additional information.
• Define the conveyor userframe Y-axis alignment as positive or negative using the dropdown menu. Y-axis alignment allows you to set the userframe with the negative Y-axis along the short axis.
• Set the orientation of the slipsheets on the station using the dropdown menu (0, 90, 180, 270).
• Additional slipsheet stations can be added using the [+] button, or deleted from the cell using the [-] button.
3.2 Understanding the EasyPallet® Pendant Interface
The EasyPallet® control panel application is a separate software application running on the embedded pc. It is intended to enable easy editing of various EasyPallet® cell definitions including; pallet patterns, layer patterns, boxes, and empty pallets directly from the embedded pc.

You will find many of the screens and interfaces to be very similar to those accessed from the EasyPallet® pendant application. While the functions may be similar, the pc interface enables more detailed control of the various component definitions as well as direct editing of the xml files.

To access the EasyPallet® control panel application, you must connect to the embedded PC operating system using a monitor, keyboard, and mouse. Make certain the monitor, keyboard and mouse are properly installed before proceeding with the following procedure (refer to chapter 2.2 "Connecting Monitor, Keyboard, and Mouse").

The EasyPallet® control panel is described below.
4.1 Main Control Panel

4.1.1 File Menu

The {Exit} shuts down the EasyPallet® application.

4.1.2 [Re-Load All Databases]

EasyPallet Control Panel reads all databases when the application is started. If the operator creates new database files or modifies existing files, then they should press the [Re-Load All Databases] button to force the EasyPallet Control Panel application to reload or refresh all database information.

The [Select Base Dir.] button allows you to switch directories to support multiple robotic palletizing cells (see chapter 2.11 “Multiple Cell Support”).

4.1.3 Project File Edit and Select Functions

Select the project you wish to work with from the dropdown menu. The [XML Ed] button is used to open the xml source files in the default xml editor.

4.1.4 Build Major Options

The Build Major Options frame allows you to build the same pallet pattern with a number of different options. These options include the tool type and associated Gripper Configuration file, building the job with a different set of job template variables, building the job on a different pallet station, or changing the job pattern number.

The {Select Tool} dropdown menu allows you to build jobs with different tool types. Gripper files are typically defined by the number of boxes and box alignment (end-to-end or side-to-side). For example:

- Vac2BoxEtoE - 2 box gripper with 8 vacuum cups. Boxes are aligned end-to-end on the gripper.
- Vac4BoxStoS - 4 box gripper with 16 vacuum cups. Boxes are aligned side-by-side on the gripper.

The gripper definitions are stored in the GripperDB.xml file. Grippers can be added, modified and updated directly in this file. Refer to chapter 6.3 “Gripper Definitions Database” for additional information on adding or modifying gripper definitions.
The {Gripper Config.} dropdown menu allows you to select the gripper configuration file to be used in the project. The [XML Ed] button is used to open the xml source files in the default xml editor.

The {Select Job Temp.} dropdown menu selects the job template used to create the robot jobs. Two sample job templates are included with EasyPallet®:

- Example2_Joint_Project template - creates jobs using the Joint coordinate system. Each robot position is recorded using the pulse counts for each manipulator axis (S,L,U,R,B,T).
- Example2_XYZ_Project template - creates relative jobs based on the XYZ (Cartesian) coordinate system. Each robot position is defined by the robot position and orientation from the origin of the Cartesian coordinate system (X, Y, Z, Xz, Yz, Zz).

Each template creates a Master job along with Pick, Place, Init, and Inc robot jobs as described below:

- Master job - calls the Pick, Place, Init, and Inc jobs as required.
- Pick job - defines how product is picked up from the conveyor.
- Place job - defines how product is placed on the pallet.
- Init job - initializes counters at the start of the process.
- Inc job - increments counters and is updated after every pick and place.

The {Pallet Number} dropdown menu defines the pallet station number in a multi-pallet cell.

The {Pattern Number} dropdown menu defines the pallet pattern to be used on this pallet station.

### 4.1.5 Build Misc. Options

The Build Misc. Options frame provides options that help determine what jobs are included in the project.

The {Create Master Job} selection allows you to elect not to update and create the Master job, updating and creating only the pick, place, init, and inc .jbi job files. This can be very useful when making minor changes to the cell without shutting down production.

The creation of robot job files is controlled by a set of xml template files. Advanced users familiar with XML programming can modify these files and create new project templates. See chapter 6 ?Advanced Programming (XML)? for more detailed information.
The \{Keep Box Orientation\} selection maintains the default box orientation.

4.1.5.1 Keep Order

The EasyPallet® software features an automatic box placement planner. This automatic planner is heavily influenced by the layer pattern and the type of gripper being used. The planner considers how many boxes the gripper can hold and their orientation (end-to-end or side-to-side as shown below).

![Diagram showing box orientation]

The planner considers the layer pattern and determines the best sequence for picking and placing boxes.

This planner works for the majority of pallet patterns using Suction cup or vacuum grippers. These grippers are the most flexible due to the fact that the surface of the gripper is always above the current box layer. This provides EasyPallet® with more options when planning the box placement sequence.

Select the Keep Order Command if you wish to override the automatic sequence planner.

When this option is selected, the EasyPallet Automatic Sequencer plans the layer pattern with the exact order provided in the Box Order Override. Compliance is guaranteed to within one pickup.

4.1.5.2 Look Ahead Jobs

The \{Look Ahead Jobs\} selection allows you to create look ahead jobs for use with the cell PLC.

4.1.6 View Commands

The View Commands frame enables you to view the EasyPallet® commands and job directory.
4.2 Pallet Pattern Editor

The Pallet Pattern Editor allows the user to define pallet builds layer by layer. Each layer is defined as a separate item that has its own unique values for each of the entries shown below.

4.2.1 Edit Functions

The Edit Functions frame allows you to save, modify, and delete pallet patterns. The [XML Ed] button opens the MasterTemp.xml file in the default xml editor. The master template can be modified and updated directly in this file (see chapter 6.1 “Job Templates”).

4.2.2 Pallet Table with Preview

The pallet table is used to define the pallet, layer by layer.

- Build each layer of the pallet using the table, adding additional layers with the [Add Layer] button.
- Select the box used with the {Box} dropdown menu (16l_6p85, box12x12x12, etc).
- Select the layer pattern using the {LayerName} dropdown menu.
- {LayerRotation} rotates the entire layer pattern (0, 90, 180, 270).
- Select a Manual Sequence file if you wish to control the exact Pick and Place sequences for this layer. Leave this selection empty if you want to use the EasyPallet Sequence planner.
- {SlipSheet Layer} defines the layer as a slipsheet. LayerName file 1.xml is typically used.
- Use the (Build Pattern on Pallet) dropdown window to define the empty pallet.

The [View Layer Pattern] button allows the user to view the results of their selections in the preview window.

This is an extremely powerful tool to use when creating new pallet patterns or experimenting with box sizes.
4.3 Layer Pattern Editor

The Layer Pattern Editor allows the user to define new layer patterns, modify existing patterns, and visually verify pattern, box, and pallet selections.

*Fig. 4-32: Layer Pattern Editor*

4.3.1 Layer Pattern - Edit and Select Functions

The "Layer Pattern -- Edit and Select Functions" frame is used to define the layer pattern, pallet, and box size. Layer patterns can be created, modified, and deleted.

- Select the layer pattern using the {Layer Pattern} dropdown menu (10G, 10E, etc).
- Select the pallet with the {Using Pallet} dropdown menu.
- Select the box with the {Using Box} dropdown menu.

The [XML Ed] button opens the LayerDB.xml file in the default xml editor. The layer pattern definitions can be modified and updated directly in this file.

4.3.2 Layer Pattern Elements

The "Layer Pattern Elements" frame is used to precisely define the layer pattern. New patterns are created using segments or groupings of boxes. These box segments are then placed relative to the previous segment until the pattern is defined.
4.3.3 Box Order Over Ride

The Box Order Over Ride allows you to override the box placement order. When fixed clamper or fixed fork style grippers are used, there are additional gripper interface issues that must be considered. Obstructions such as walls, support columns, or other pallets may require the robot to place the boxes in a different order than the one calculated by the EasyPallet® automatic planner.

In these cases, the user has a number of tools available to modify the box placement order or sequence. These tools include the Box Over Ride Feature, the Keep Box Orient. (Orientation), and the Keep Order commands. When used together properly, these functions allow the user to modify the box order to meet the desired sequence.

Example 1: Layer Setup with Fixed Fork Gripper

This example uses the 13 Box Pattern layer pattern, and the Fixed Fork, 3 box, end-to-end gripper.

{Box on Pallet Overhang}

The “Box on Pallet Overhang” frame defines the amount a box pattern can overhang the pallet in the X and Y directions.

{Box Air Gaps}

The “Box Air Gaps” frame defines the gap between boxes in the X, Y, and Z directions.

{Show Box Coordinates}

The “Show Box Coordinates” frame displays the box coordinates on the pallet.
To create this pattern with a 3 box Fixed Fork gripper, we want the Asynchronous placement to occur first. This is because the Fixed Fork gripper must have additional room around the boxes as they are placed individually. On the EasyPallet Control Panel:

1. From the EasyPallet® Control Panel, add the 13Test1 layer pattern to a pallet pattern.
2. Select the following Box Order (see Figure below).
3. Select the following build options (see Figure below).
   a) Keep Box Orient.
   b) Keep Order
4. Create your palletizing jobs.
Table 4-2: Placement Sequence on Pallet

<table>
<thead>
<tr>
<th>Place Box #</th>
<th>Gripper Position on Pallet</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td><img src="154652-1CD" alt="Image" /></td>
</tr>
<tr>
<td>12</td>
<td><img src="154652-1CD" alt="Image" /></td>
</tr>
<tr>
<td>13</td>
<td><img src="154652-1CD" alt="Image" /></td>
</tr>
<tr>
<td>10, 9, 8</td>
<td><img src="154652-1CD" alt="Image" /></td>
</tr>
<tr>
<td>6, 7</td>
<td><img src="154652-1CD" alt="Image" /></td>
</tr>
<tr>
<td>3, 4, 5</td>
<td><img src="154652-1CD" alt="Image" /></td>
</tr>
<tr>
<td>1, 2</td>
<td><img src="154652-1CD" alt="Image" /></td>
</tr>
</tbody>
</table>
4.3.4 Layer Pattern Table with Preview

The layer pattern table is used to precisely define the layer pattern segment by segment. Each portion of the pattern is placed using the table, adding additional segments as required.

{Rows}
The {Rows} cell defines the rows of boxes for a segment.

{Cols}
The {Cols} cell defines the columns of boxes for a segment.

{Rot.}
The {Rot.} cell defines the rotation of a segment (0, 90, 180, 270).

{Spread}
The {Spread} cell spaces the boxes of a segment evenly across the length or width of a previous segment, adding spacing as necessary.

{Seg. Align}
The {Seg. Align} cell defines the segment alignment in relation to the previous segment.

{Ext. Y}
The {Ext. Y} cell places the new segment in the Y axis relative to the previous segment when set to True. When set to false, the segment is placed in the X axis.

{Ext. Neg}
The {Ext. Neg} cell places the segment in the negative X or Y axis relative to the previous segment.

[Delete]
The [Delete] button removes the segment from the pattern.

Layer Pattern Table with Preview is an extremely powerful tool to use when creating new layer patterns, or when experimenting with box sizes on existing patterns. The preview provides an accurate depiction of how your pallet pattern will look with different box sizes.
4.4 Box Editor (PC Based)

The Box Editor provides a graphic interface to the BoxDB.xml database. Users can modify existing box definitions or enter new box configurations.

4.4.1 XML Editing Tools

The XML Editing Tools frame provides controls to save changes to the BoxDB.xml database, re-load the existing database without saving changes, and open the BoxDB.xml file in the default xml editor for direct editing. For detailed information on the BoxDB.xml file, refer to chapter 6.2 "Box Definition Database".

[Save]

The [Save] button saves changes made in the Box Editor directly to the BoxDB.xml database.

[Re-Load Databases]

The [Re-Load Databases] button clears any changes made in the box editor and reloads the existing BoxDB.xml data.

[Box XML Editor]

The [Box XML Editor] opens the BoxDB.xml file in the default xml editor. For detailed information on the BoxDB.xml file, refer to chapter 6.2 "Box Definition Database".

4.4.2 Display Tools

The Display Tools frame enables the user to view box definitions in either millimeters or inches using the available radio buttons.

4.4.3 Box Definition Table

The Box Definition Table displays the data for every box definition recorded in the BoxDB.xml file. Box definitions can be modified directly in the table, new box definitions added, and existing box definitions deleted.
4.5 Empty Pallet Editor

The Empty Pallet Editor provides a graphic interface to the PalletDB.xml database. Users can modify existing pallet definitions or enter new pallet configurations.

### 4.5.1 Edit Tools

The Edit Tools frame provides controls to save changes to the PalletDB.xml database, re-load the existing database without saving changes, and open the PalletDB.xml file in the default xml editor for direct editing. For detailed information on the PalletDB.xml file, refer to chapter 6.4 "Pallet Definition Database".

[Save]  
The [Save] button saves changes made in the Empty Pallet Editor directly to the PalletDB.xml database.

[Re-Load Databases]  
The [Re-Load Databases] button clears any changes made in the Empty Pallet Editor and reloads the existing PalletDB.xml data.

[Empty Pallet XML Edit]  
The [Empty Pallet XML Edit] opens the PalletDB.xml file in the default xml editor. For detailed information on the PalletDB.xml file, refer to chapter 6.4 "Pallet Definition Database".

### 4.5.2 Display Tools

The Display Tools frame enables the user to view empty pallet definitions in either millimeters or inches using the available radio buttons.

### 4.5.3 Empty Pallet Definition Table

The empty pallet definition table displays the data for every pallet definition recorded in the PalletDB.xml file. Empty pallet definitions can be modified directly in the table, new pallet definitions added, and existing pallet definitions deleted.
The Batch Build window allows you to create multiple sets of palletizing jobs with one easy step. With Batch Build, you can create any number of batch files. For example; you can create a batch for each cell, a batch for each pattern, or a batch of all patterns used for a particular pallet. Batch Build can also be used for trouble shooting. By creating a batch containing the worst case reach patterns, you can test reach solutions across multiple build combinations.

**Fig. 4-33: Batch Build Tab**

### 4.6.1 Batch Build Edit and Save Functions

The “Batch Build Edit and Save Functions” frame is used in conjunction with the table to define each job build included in the batch. Job batches can be created, modified, and deleted. Each job included in the batch is defined using the dropdown windows in each variable column.

The [XML Ed] button opens the BatchDB.xml file in the default xml editor. The job batch list can be modified and updated directly in this file.

### 4.6.2 Batch Build Control Panel

Once a batch has been defined and saved, you can start the batch job creation by pressing the [Start Batch] button. EasyPallet® keeps track of the number of the total number of builds, as well as the number of good and bad builds. The batch process can be stopped at anytime by pressing the [Abort Batch] button.
4.7 Manual Sequencer

The Manual Sequencer tab allows you to build pallet patterns using any pick and place sequence desired. This function is independent of box and pallet size, automatically stretching and shrinking as the box and pallet sizes grow larger or smaller. The manual sequencer is a great time saver, allowing you to reuse previously created manual sequence files over and over again.

The creation of manual sequence files is also very easy as most math calculations have been removed from the file creation process. The visual process uses existing EasyPallet® generic layer patterns and gripper pickup locations to automatically determine the correct gripper placement locations.

4.7.1 Layer Manual Sequence File -- Edit and Select Functions

Select a manual sequence file directly from the dropdown window or click the [SaveAs] button to create a new manual sequence file. The files can be edited directly in your default XML editor by pressing the [XML Ed] button.

**Layer Manual Sequence File Naming Conventions**

EasyPallet Manual Sequence files adhere to the following naming convention:  XXX_MSF_Example_YPi_ZPL

- XXX is the associated Layer Pallet Pattern
- Y is an integer and gives the number of pickup operations the robot would make in this Manual Sequence operation
- Z is an integer and gives the number of place operations the robot would make in this Manual Sequence operation

4.7.2 Layer Info Tab

The Layer Information tab allows you to quickly select the basic Manual Sequence file information. This information must be setup before defining or refining the pickup and placement operations.
Layer Pattern Selection

Use the Layer Pattern drop-down menu to select the desired layer pattern that you want to build with these custom pick and place operations. Please note that these are selected from among the standard 200 plus layer placement patterns supplied with EasyPallet.

![Layer Pattern Selection](image)

**WARNING**

The Layer Pattern must be set up before the Manual Sequence file is created. Further, if the Layer Pallet is modified after the creation of the Manual Sequence file, it may have to be recreated from scratch.

Tool Type Selection

Use the Tool Type drop-down menu to select the generic type of gripper used to build this pallet layer. Select the [Update Tool Type] button to make the new tool type active and update the gripper and gripper number of zones selection boxes.

Using Pallet Selection

Pallet selection is labeled as Using Pallet. This is a reminder that the Manual Sequence file is independent of actual pallet size.

Using Box Selection

Box selection is labeled as Using Pallet. This is to remind the user that the Manual Sequence File is independent of actual box size.

Refresh

Use the [Refresh] button to view the basic Layer Pattern that you have chosen.

Box and Pallet selections are not saved as part of the Manual Sequence files. However, they must be selected before the Layer Pattern can be properly displayed on the LayerInfo tab.

4.7.3 Pick and Place Editor

The Pick and Place Editor allows you to define exactly which boxes are picked up as well as the placement sequence.
Pick Horizontal Scroll Bar

The Pick horizontal scroll bar allows you to view a summary of the boxes that are picked for the selected operation.

Pick#1
This example shows that Pick#1 picks up and places boxes 7 and 8 in one operation.

Pick#2
Pick#2 picks up 4 boxes (box numbers 3, 4, 5, & 6) and places them in a single drop.

Pick#3
Pick#3 picks up boxes 1 and 2 and places them in a single drop.
Place Horizontal Scroll Bar

The Place horizontal scroll bar provides a summary of the boxes that are placed in this operation.

Place operations are associated with a specific Pick operation.

A-synchronous or split drops are shown as separate operations. In the following example, boxes 3 and 4 are picked up. Box 4 is placed, followed by the placement of box 3.

Pick Boxes 3, 4, Place #1, Box 4 (Pickup View)

Pick Boxes 3, 4, Place #1, Box 4 (Place View)
Adding Pickup and Place Sequences

The [Add] button opens the Add Pick & Place window allowing you to create new pick and place sequences. The user enters the box numbers he wishes to pickup. Each box number should be separated by a comma. Up to 4 Place operations can be defined for each pickup operation. The following example shows a pickup of boxes 1 to 6 and a single place operation of boxes 1 to 6.
Add Boxes Insert Location

Where the new Pickup and Place Sequence is added is dependent upon where the Pickup Horizontal Selection Scroll bar is located. If it is located at the last pickup operation, the new Pick/Place is added at the end of the queue. Otherwise, the new Pick/Place sequence is added before the currently displayed Pickup operation.

Delete Button Operation

The [Delete] button deletes the current PickUp sequence. This includes selecting the required Pickup pattern, setting the Gripper Pickup TCP location, determining the correct Pickup vectoring, and selecting which gripper zones to activate or turn ON.

4.7.4 Pick Tab

The Pick tab allows you to setup or modify the pick sequence. This includes selecting the required Pickup Pattern, setting the Gripper Pickup TCP location, determining the correct Pickup vectoring, and selecting which gripper zones to activate or turn on.
Pick Pattern Selection

The Pick Pat. drop down menu supplies a list of available pick patterns. Pick patterns provide the number and formation of the boxes that will be picked up by the gripper. You can create as many additional pick patterns as desired.

Once the pick pattern is selected, place the gripper TCP at the desired location to where box pickup will occur.

Setting Up the Gripper TCP

This step allows defines where you want to attach the gripper TCP to the boxes. The following picture shows the gripper TCP (center of the red circle), and the gripper X and Y axes.

Gripper Pickup Length Units

You can change the gripper pickup length units by selecting either percent or millimeter length units. Percent is the default unit. In most cases the gripper TCP is placed in the center of the pickup group. The center of the pickup group is 50% of the pickup group’s maximum X and Y values. Even as the boxes and conveyors change size, the 50% value remains a constant.

Regardless of the selected gripper TCP entry units, the values stored in the Gripper XML Configuration file are always converted to and stored as a percentage as shown below.

Pick and Place Vectoring

Vectoring is the process by which the gripper approaches the boxes on the conveyor (Pickup) or drops boxes on a pallet with a sliding motion. Vectoring allows the gripper to slide into the correct location without box to box collisions or edge clipping. Vectoring is required in some types of palletizing operations because small placement errors, pickup TCP errors, box size variations or box on gripper oscillations can cause boxes edges to clip one another.

Vectoring works the same for box pickup and placement operations. Vectoring is setup visually by incrementing and decrementing the Vector Horizontal Scroll Bar. All Vectoring is specified relative to the gripper axis system.
WARNING

It is the users responsibility to ensure that vectoring is setup in a way that will not cause crashes.

Vector directions are shown in both the Pick and Place Vector displays directly above the horizontal scroll bar as an orange arrow. It is also shown in the main Pick and Place layer display as a orange line.

Table XXXX Manual Sequence File 8D_MSF_Example_4Pi_6Pl Placement Vectoring Example

Table 4-3: Placement Vectoring

<table>
<thead>
<tr>
<th>Vectoring Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Vectoring</td>
<td></td>
</tr>
<tr>
<td>270° Vector (relative to Tool X axis)</td>
<td></td>
</tr>
</tbody>
</table>
Pick and Place Gripper Zone Control

Gripper zones are controlled by the GripperZones check boxes in both the Pick and Place Tabs. When gripper zones are ON, they are displayed as a solid color. When they are OFF, they are translucent.

**WARNING**

It is the user's responsibility to setup and ensure that the correct gripper zones are used and that boxes are picked up and dropped at the correct location to avoid robot crashes.

The following pick operation example depicts a two zone gripper with Zone 1 Off and Zone 2 On.
Gripper Zone Control Numbering

Gripper zones always start with the most negative X-axis value being Zone 1 and zone numbers increase as X increases. See the example below.

4.7.5 Place Tab

The Place tab provides options for box placement including gripper TCP position, gripper zones used, as well as the place vector (approach vector).
Display options allow you to change the display scale as well as how the box and gripper coordinates are displayed.

**TCP Place Location**

Defining a place operation is generally easier than defining a pick operation because the pickup location, gripper type, and layer pattern automatically determines one of two placement locations that differ by a gripper rotation of 180 degrees.

**Gripper Placement Vectoring**

Gripper placement vectoring behaves similar to Pick and Place Vectoring.

**Gripper Zones Operation**

Gripper zone operation behaves similar to the Pick and Place Gripper Zone Control.
EasyPallet® is an incredibly flexible and versatile tool. There are many ways to accomplish the same task depending on your preferences. Pallet jobs can be created, modified, and deleted working solely from the programming pendant, or the same work can be accomplished directly from the embedded pc using a monitor, keyboard, and mouse.

Some situations may warrant creating pallet, box, and pattern definitions directly from the embedded pc using the EasyPallet® Control Panel. For other situations it may be easier to modify definitions directly in the xml files while creating the actual jobs using the programming pendant. Regardless of how you interface with EasyPallet®, there are some basic steps that must be performed that are independent of the way you interface with EasyPallet®.

5.1 Overview

While limited stand-alone operation is possible, EasyPallet® jobs are designed to work with a cell controller PLC (Programmable Logic Controller).

Typically, the PLC controls the workcell and is responsible for determining the operational status of the cell. It is responsible for controlling, monitoring, and planning the throughput of the cell. The PLC determines which conveyor to pick from and which pallet station to place the product.

The Master job created by EasyPallet® acts as a slave to the cell controller PLC. The Master job cycles through the robot I/O groups waiting for a Pick and Place command.

Once a command is received, the Master job calls the correct pick and place jobs for the specific pallet pattern (dictated by the PLC) being placed on one of the cell's pallet stations.

*Fig. 5-34: EasyPallet® - PLC Relationship*

The EasyPallet® jobs keep track of each pallet's status. Status includes:

- Current layer for each pallet
- Current height of the pallet
- Number of layers the pattern contains
- Number of picks per layer
- Current pick for the layer
- Current robot status (idle, picking or placing boxes).
EasyPallet® uses a set of robot variables (typically B-variables) to keep track of each pallet's state. Advanced users can use these variables to implement a number of optional job functions and features for their specific work cell.

These optional functions and features are required because there are no standard methods for implementing many of the required commands necessary for custom palletizing cells. These functions also require a certain amount of manual programming to implement. For example; if the robot misses a pick because a box is damaged, how is this situation to be corrected? A number of options exist, including:

- Attempt to re-pick the boxes
- Stop the system and manually clear the layer
- Stop the system and manually complete the layer

Each of these solutions are implemented in a different way. EasyPallet® programs work with any of these solutions.

5.2 Process Flow

EasyPallet® runs on two separate processors (embedded pc and programming pendant) that communicate via an FTP link (File Transfer Protocol). XML files and databases are used to store required information. As shown below, a subset of the databases are duplicated and stored locally on the compact flash (CF) card.
5.2.1 File Synchronization

For the system to operate properly, the databases in both locations (Embedded PC and Pendant CF Card) must remain synchronized. During normal operation, changes and commands requested by the operator via the pendant, will result in local database files being updated and sent to the embedded PC. The EasyPallet Server, on the embedded PC, replaces the existing files with the updated files from the pendant and file synchronization is maintained.

When creating or modifying files on the embedded PC, the user is responsible for making sure that these newer files are transferred to the programming pendant via the CF card. The best way to do this is with a standard CF card reader attached to the embedded PC. Files can then be dragged and dropped into the CF card.

5.3 Embedded PC Directory Structure

5.3.1 Database and Application Directories

The following diagrams show the directory structure for EasyPallet®. The main hub file labeled "EasyPalletPC2" can be found in the embedded PC at: C:\Program Files\Motoman\EasyPalletPC2.

Fig. 5-35: EasyPallet® Directory Structure

The lines, leading from the hub, identify the subdirectories where the EasyPallet® application and database files are located. For instance, the DB_Box line, identifies the subdirectory where the box database is stored on the embedded PC. So the Box DB is found in the following directory:

C:\Program Files\Motoman\EasyPalletPC2\DB_Box

EasyPallet® directories adhere to the following naming conventions:

- **DB_XXX** directories contain one xml database file.
- **Dir_XXX** directories contain multiple xml files that each define and independent set of data. The Dir_LayerPatterns directory follows this pattern.
- **EXE_XXX** directories contain the EasyPallet application executable file with all of its required software components or *.dll files.
- **FTPxxxxx** directories contain various commands or *.epc files and database files that are being moved between the Pendant and PC.
\EasyPalletPC2\ DB_Box > BoxDB.xml - This database contains the definitions of all boxes in the system.

\EasyPalletPC2\ DB_MotoSimCell > DBMotoSimCellDB.xml - This database points to the cell name and path. For robot on a rail systems, it also contains a description of the rail including maximum length and maximum pulse counts.

\EasyPalletPC2\ Dir_FTPSettings - This directory is used to set up the FTP Link parameters. The currentFTPFile.xml file points to the pendant FTP parameters in the FTPNXPendant.xml file.

In test situations, the currentFTPFile.xml file can be pointed to the FTPFilezilla.xml file. In this configuration the EasyPallet server monitors local files in the C:\DiskOnChip\PPallet\FTP_Send and C:\DiskOnChip\PPallet\FTP_Receive directories. This configuration can be used to test EasyPallet Server operation when a controller or pendant is not available.

\EasyPalletPC2\ Dir_PalletPatterns - This directory contains all the pallet pattern defined in your palletizing systems. EasyPallet Pattern definitions contains all the information required to create a complete pallet such as the number of layers, box by layer, layer pattern rotations etc.

\EasyPalletPC2\ FTPCommandsFromPendant - The FTP FileZilla server moves files from the pendant Send directory, on the Pendant CF card (\PPallet\FTP_Send) to this directory.

\EasyPalletPC2\ FTPCommandsBackupPendant - When the FileZilla Server moves command and data files from the pendant to the PC it also tells Easy Pallet Server that the files have arrived. EasyPallet® then moves the files from the \EasyPalletPC2\ FTPCommandsFromPendant to this directory.

\EasyPalletPC2\ FTPCommandsBackupFromPC - When EasyPallet® is preparing data to be moved the pendant it copies all data to this file. Once the data is ready to be sent to the pendant, the EasyPallet server moves the files from here to the \EasyPalletPC2\ FTPCommandsFromPC, the PC-ready-to-send directory.

\EasyPalletPC2\ FTPCommandsFromPC - Files placed in this directory are sent to the Pendant CF card (\PPallet\FTP_Receive) directory.

\EasyPalletPC2\ Dir_BatchBuilds - This directory contains all the Batch Build XML files the user has created.

\EasyPalletPC2\ Dir_LayerSeqFiles - This directory contains all the custom manual layer sequence files that the user has created.

\EasyPalletPC2\ Dir_AIIXMLschemas - This directory contains all XML schema files. These files are used by EasyPallet to determine the proper form of the various XML database files. Do not move, modify, or delete these files.

\EasyPalletPC2\ CustomTemplates\MotoPalletBased - This directory contains a number of subdirectories. Each subdirectory contains a set of XML template Files.

The following figure shows the expanded directory:

C:\Program Files\Motoman\EasyPalletPC2\CustomTemplates\MotoPalletBased\Joint_IOControl_SLOW

This directory contains all the template files required for the creation of one set of palletizing jobs.
Fig. 5-36: Custom Templates Directory

\EasyPalletPC2\ DB_Grippers - GripperDB.xml - This xml file contains a description of each type of gripper used in EasyPallet®.

\EasyPalletPC2\ DB_Pallet - PalletDB.xml - This file contains a listing of each type of empty pallet that can be used in your system.

\EasyPalletPC2\ Dir_EasyPalletProjects - This directory contains a number of xml Project database files. These files:

- List a number of build options used on the last build,
- Describes each Pallet conveyor and slipsheet stations used in the current cell file,
- Lists the various I/O blocks that can be used to transfer data between the PLC and the Easy Pallet programs.

\EasyPalletPC2\ Dir_LayerPatterns - This directory contains over 200 generic layer patterns and a *.bmp representative picture for each pattern.

\EasyPalletPC2\ EXT_EasyPalletServer2 - This directory contains the EasyPalletServer.exe file and associated software components.

\EasyPalletPC2\ EXE_EasyPalletPCControlPanel - This directory contains the PalletizingForm.exe file and all associated software components.
5.3.2 Embedded PC - Cell and Job Archive Directories

Fig. 5-37: Cell and Job Directories

The previous figure shows the root of the cell data. The directory labeled EPT1 and highlighted in yellow, is the name of the cell file specified in the C:\ProgramFiles\Motoman\EasyPalletPC2\DB_MotoSimCell\MotoSimCell DB.xml file. In this example the hub is pointing to: C:\Program Files\Motoman\MotoSim EG\Cells\EPT1.

5.3.3 Embedded PC - Robot Subdirectory

Within the main directory is an important subdirectory called the Robot type subdirectory. In this example, the robot type subdirectory is named EPL160-A10. This directory points to the embedded PC directory. In this example it points to:

C:\Program Files\Motoman\MotoSim EG\Cells\EPT1

This is the subdirectory where robot jobs are converted from the intermediate INF File format to the robot job or *.jbi file format. The robot subdirectory is also the location of the robot UFrame.cnd, Tool.cnd, and All.prm files. This subdirectory also contains error files generated during the INF to JBI conversion process. These files are named: XXX_INFTOJBI_ERROR.TXT (where XXX is the job name).

The predominate error shown in these files are reach errors.
5.3.4 Embedded PC - Job and Command Archive Files

The Custom Templates subdirectory, highlighted in green in fig. 5-37, contains subdirectories for each relevant job. Each time a build request is issued, a subdirectory for all relevant job information is created. In the example above, the MasterJ2 subdirectory is the repository for that build.

These archive directories serve two major purposes: to act as a backup and as a troubleshooting aid. The most important subdirectories and files are explained below: \xxxxxx\ is an identifier for the last job build.

\Custom Templates\xxxxxx\ASO - This directory contains the pick and place sequence files *.asi and *.aso files for the build.

\Custom Templates\xxxxxx\BuildLog - This directory contains the BuildLog.xml file. This file shows major steps and errors that have occurred in the job creation process.
5.4 Pendant CF Card Directory Structure

The following diagram shows the major files and directories on the EasyPallet Pendant CF Card. The yellow hub is the root directory of the CF card. The green files show the EasyPallet database directories. These directories hold copies of the Easy Pallet XML databases. When a change is made to these files, that file is downloaded to the embedded PC so the databases remain synchronized. The light purple files are the FTP send and receive directories that are monitored by the embedded PC Filezilla FTP Server.

Tan files are only used during the initial installation of the EasyPallet® application on the Pendant.
5.5 EasyPallet® Conversion Overview

EasyPallet® not only makes it easy to create robotic palletizing jobs, but the jobs created also have a high degree of flexibility in form and function. To this end, the EasyPallet® creates a number of reusable databases and files. When a build is requested, the Easy Pallet server, selects, loads and manipulates a large amount of data. As you can see below, at least 4 conversions are used to turn the data into a set of palletizing jobs.

While not the most straight forward method, it encourages maximum reuse of a large number of software files and components. It is also a fast and easy way to create a large number of jobs very quickly.

While these conversions are typically invisible to the end user during normal job creation, a basic understanding of the conversion process will certainly help troubleshooting should an error occur.
Conversion 1 - Starts when the user selects the [Create Jobs] command and ends with the output of *.ASI and *.ASO files.

The ASI files are a key input to the various Pick/Place Sequence planners. The ASO files are the output from the Sequence planners.

The *.ASI file provides the Gripper type, Box size and box location (box center x, y and rotation) on the pallet. The *.ASO files provide a list of Box Pick and Place orders. The ASO file is a primary input required for the creation of the XML Job files.

These files are located stored on the embedded PC at the following directory:

`\Custom Templates\XXXXX \ ASO - where XXXXX is the identifier of the last job build`

Conversion 2 - During this conversion, each pickup and placement location is merged with job template files. The template files eventually create the robot job files.

Located in the job template files are $command and $locations. During the conversion process these commands are replaced with actual location data and operations. The templates are used to add robot approach and departure moves to the initial pickup and placement locations. Additional commands are also inserted into the jobs during this conversion.

Major inputs to this process are the *.ASO files, Template Files, and some of the Robot Cell data. The output of this conversion is a set of XML Robot jobs. These jobs are identical to the Points Importer XML job format. These files are located stored on the embedded PC at the following directory:

`\Custom Templates\XXXXX \ XMLJobs - where XXXXX is the identifier of the last job build`

Conversion 3 - During this conversion the robot XML job files are converted into robot .inf files.

Conversion 4 - In this conversion, INF files are converted to robot JBIs. During this conversion, robot reach is checked and joint limits may be exposed. Problems during this conversion process generally result in an error file being generated. The coordinate system used in the jobs may either be robot joint space or robot User Frame coordinates. These files are located stored on the embedded PC at the following directory:

`\Custom Templates\XXXXX \ JBIJobs - where XXXXX is the identifier of the last job build`

5.5.1 Master Job Structure

EasyPallet creates two types of jobs. Jobs with alphabetic names are part of the master job structure. These jobs do not change as pallet station or pattern changes. Numbered jobs are part of the variable job structure. These jobs change as the selected pallet station and patterns are changed.

5.6 Robot on a Rail Applications

For some palletizing systems, the robot may need to be placed on a rail.

To setup a Robot on a Rail system:

1. Setup Robot Cell.
5 Operation
5.6 Robot on a Rail Applications

a) Load All.prm file from a robot that has 7 linear axes defined. This all.prm file must be placed in the robot subdirectory of the Cell file.

b) Create User Frames at each Pallet and Conveyor Location.

2. Add the "Static or Dynamic Rail Option" commands to the MotoSimCellDB.xml file (C:\Program Files\Motoman\EasyPalletPC2\DB_MotoSimCell\MotoSimCellDB.xml).

3. Create palletizing jobs.

The robot on a rail may be setup as either a static or dynamic rail option. With the static rail option, the robot moves to a static rail location and then works in a user frame that is based on that rail position. No further rail movement is employed in the transformed robot job. Use this option when the robot needs to move to a point on the rail and then act as a robot only system.

5.6.1 Robot on a Rail Setup - Static Rail Option

Add the following command to the MotoSimCellDB.xml.

<ExternalAxes AxesPulsePosition="15000" AxesType="Base" /></ExternalAxes>

The user frame is relative to the rail location (i.e. if you move the rail location, the user frame location moves).

The Dynamic rail option places the robot axis that is aligned with the rail position at a set location and performs rail movement whenever any change in the rail aligned axis is required. Use this option when the robot needs to work at many points on the rail.

5.6.1 Robot on a Rail Setup - Static Rail Option

Add the following command to the MotoSimCellDB.xml.

<ExternalAxes AxesPulsePosition="15000" AxesType="Base" /></ExternalAxes>

The user frame is relative to the rail location (i.e. if you move the rail location, the user frame location moves).
AxesPulsePosition - This is the static rail location (in pulse counts) that the robot moves. Once the robot moves to this location, it acts as a 6-axis robot system only.

5.6.2 Robot on a Rail Setup - Dynamic Rail Option

BSMaximumLength - gives the robot maximum length in millimeters.

BSMaximumCounts - gives the encoder counts when the rail is placed at its maximum length position.

BSUFAxisAlignedWithBase - defines the Robot Axis that is aligned with the rail length.

BSUFNumber - gives the robot user frame that is setup and aligned with the robot rail system.

BSRobotBaseAxis = BSUFAxisAlignedWithBase

BSRobotMinBaseAxisValue is not used and should be set to 0.0.

BSRobotMaxAxisValue gives the value of the robot axis that is aligned with the rail system.

The BSRobotMaxAxisValue can be modified for different robot models or to adjust the G-Code conversion process. For example, the X value can be increased to extend the robot arm and thus lower the robot elbow. It can also be decreased to bring the tool closer to the robot base, making the process run more smoothly.

<BaseStation>
  <BSMaximumLength>15076.488</BSMaximumLength>
  <BSMaximumCounts>1235115</BSMaximumCounts>
  <BSUserFrame>
    <BSUFAxisAlignedWithBase>X</BSUFAxisAlignedWithBase>
    <BSUFNumber>1</BSUFNumber>
  </BSUserFrame>
  <BSRobot>
    <BSRobotBaseAxis>X</BSRobotBaseAxis>
    <BSRobotMinBaseAxisValue>0.0</BSRobotMinBaseAxisValue>
    <BSRobotMaxAxisValue>700.0</BSRobotMaxAxisValue>
  </BSRobot>
</BaseStation>

The robot axis that is aligned with the rail is limited to the minimum and maximum values stored in elements below. In this example, the robot minimum and maximum would be between 0 and 700 mm (offsets from this value are added to the rail system).

These values may need to be adjusted to maximize the reach you need or to adjust the relative position of the robot to the pallet.
5.7 Creating New Jobs Using the Pendant Application

1. Using the {Project} dropdown menu in the Project Configuration area, select the job template you wish to use to create your robot job files. Job templates determine the coordinate system (Joint or Cartesian) and job structure and can be modified to meet your specific needs. See chapter 6 "Advanced Programming (XML)", for more detailed information.

2. Select the pallet station you are creating the job for from the {Pallet #} dropdown menu.

3. Select the job pattern number you wish to use from the {Job Pattern #} dropdown menu.

4. Select the gripper being used from the Tool dropdown menu.

5. Click the [Cell Setup] button. The cell setup interface appears.
6. Using the cell setup interface, define your work cell including conveyors, pallet stations, and slipsheet stations. Refer to chapter 3.2.3 “Cell Setup User Interface” for additional information.

7. Click [Return] to return to the main screen.

8. Begin building pallet layers using the Pallet Configuration area. The {Pallet Pattern} dropdown menu can be used to load existing pallet patterns for use and modification.

   a) Starting with Layer 1 as the bottom layer, select the {Box Name} and {Layer Pattern} from the dropdown menus.

   • The [Box Editor] can be used to modify existing box definitions or add new boxes for use with EasyPallet®. See chapter 3.2.2 “Box Editor” for more information.

   • The [?] button can be used to visually review layer patterns to refresh your memory regarding their configuration. However, please remember, these pattern depictions do not represent your box dimensions. Your pattern may look very different depending on the dimensions of the boxes you are using.

   b) Continue building pallet layers using the [Add] and [Remove Layer] buttons and the {Box Name} and {Layer Pattern} dropdown menus.

9. Select the pallet being used for this job using the {Pallet} dropdown menu.

10. Name your new pallet configuration by clicking the [Edit] button. The on screen, virtual keyboard appears.

11. Using the on screen keyboard, enter the name of the pallet configuration and press Enter.

12. Once you have defined your project including box, tool, and pallet configuration, click [Save] to save the configuration under the current project name, or [Save As] to save the configuration under a new project name.

13. To create new jobs based on your new configuration, click the [Create Jobs] button. The “Send Build Request - waiting for job files from server…” message appears in the Status window. Files are sent from the programming pendant to the embedded pc and the simulation software begins processing the job request.

14. When the process is complete, “Last Action: saved requested files” appears in the status window and the set of .jbi robot job files are uploaded to the programming pendant and are available for use.

   Errors can occur in the job build process for a variety of reasons. The most common error (Error #9 Build Error) occurs due to reach issues with the robot.

   Remember to use the numbered dropdown menu to determine which middle layer is displayed when defining more than three pallet layers.
5.8 Modifying Jobs Using the Pendant Application

1. Using the {Project} dropdown menu in the Project Configuration area, select the project you wish to modify. The main project file acts as a job template determining the coordinate system (Joint or Cartesian) and job structure and can be modified to meet your specific needs. See chapter 6 "Advanced Programming (XML)" for more detailed information.

2. Change the pallet station you want to create the job for using the {Pallet #} dropdown menu.

3. Change the job pattern number using the {Job Pattern #} dropdown menu.

4. Change the gripper using the {Tool} dropdown menu.

5. Click the [Cell Setup] button to modify the basic structure of your workcell including conveyor, pallet stations, and slipsheet dispensers.

6. Select the pallet pattern using the {Pallet Pattern} dropdown menu in the Pallet Configuration area.

7. Click [Edit] to modify the pallet pattern name. Enter the new pallet pattern name using the on screen keyboard.

8. Modify pallet layers using the [Add Layer] and [Remove Layer] buttons and the {Box Name} and {Layer Pattern} dropdown menus.

9. The Box Editor can be used to modify existing box definitions or add new boxes for use with EasyPallet®. See chapter 3.2.2 "Box Editor" for more information.

10. Once you have modified your pallet configuration, click the [Create Jobs] button. The “Send Build Request - waiting for job files from server...” message appears in the Status window. Files are sent from the programming pendant to the embedded pc and the simulation software begins processing the job request.

5.9 Easy Pallet Pick and Place Sequence Options

Robotic palletizing almost always pushes the limit of robot reach and cycle times. Additional restrictions can occur due to existing cell layout or space restrictions such as working around building support columns or walls, or low ceilings. Still other obstructions include other pallets and conveyor stations. Additionally, as a robot’s payload increases, it tends to carry more and more boxes per pick and place. This makes the grippers larger and places additional restrictions on the gripper movement.

This environment requires a comprehensive robotic palletizing solution that can be very automated when the conditions allow and very flexible when the conditions demand it.

EasyPallet has three basic pick and place sequence planners that allow it to work effectively in all the situations discussed above. These planners are also gripper dependant. The 3 basic planners are called the Auto Sequencer- One Box Gripper, Auto Sequencer - Multi Box Gripper, and the Manual Sequence.

As the pyramid below shows, the full Auto Sequencer is the easiest but the most rigid, while the Manual Sequencer is the most difficult but also the most flexible. Users should always use the Full Auto Sequencer when possible and select other options as required.
The following figure and table outline when the various Sequence options might be used.

<table>
<thead>
<tr>
<th>Gripper Type</th>
<th>Easy Pallet Sequence Method</th>
<th>Characteristics</th>
<th>One Box/One Gripper Zone Rule</th>
<th>Place Vector</th>
<th>Gripper Zone Control</th>
<th>Resulting Pick/Place Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>One Box Gripper</td>
<td>Auto Sequencer (See Figure 5-21 below)</td>
<td>1. Layer Editor assigned box number are used to determine robot pick and place order. 2. Easiest to create 3. Poor Sequence Control</td>
<td>Yes</td>
<td>Auto</td>
<td>Auto</td>
<td>1,2,3,5,5,6,7,8</td>
</tr>
<tr>
<td>One Box Gripper</td>
<td>Auto Sequencer with Box Order Override Option (See Figure 5-22)</td>
<td>1. Use can create any desired order that he wishes to use. 2. Easy to create 3. Good Sequence Control</td>
<td>Yes</td>
<td>Auto</td>
<td>Auto</td>
<td>7,8,3,4,5,6,1,2</td>
</tr>
</tbody>
</table>
### 5.9 Easy Pallet Pick and Place Sequence Options

<table>
<thead>
<tr>
<th>Gripper Type</th>
<th>Easy Pallet Sequence Method</th>
<th>Characteristics</th>
<th>One Box/One Gripper Zone Rule</th>
<th>Place Vector</th>
<th>Gripper Zone Control</th>
<th>Resulting Pick/Place Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi Box Gripper</td>
<td>Auto Sequencer</td>
<td>EasyPallet Auto Sequencer examining the gripper type and the layer Pattern. It automatically plans all pick and place operations. Generally it builds from the middle out pallet out in all directions. 1. Best for Vacuum Style Grippers 2. Easiest to create 3. Poor Sequence Control 4. Sequence changes with Gripper type</td>
<td>Yes</td>
<td>Auto</td>
<td>Auto</td>
<td>(With a 2 box gripper) 3 &amp; 4 5 &amp; 6 7 &amp; 8 1 &amp; 2</td>
</tr>
<tr>
<td>Multi Box Gripper</td>
<td>Auto Sequencer with Keep Order Cmd</td>
<td>1. Generally builds from lower box number to higher box numbers 2. Easy to create 3. Average Sequence Control 4. Sequence changes with Gripper type</td>
<td>Yes</td>
<td>Auto</td>
<td>Auto</td>
<td>1 &amp; 2 3 &amp; 4 5 &amp; 6 7 &amp; 8</td>
</tr>
<tr>
<td>Multi Box Gripper</td>
<td>Manual Sequencer</td>
<td>Provides the most flexibility. Here the user can create any pick and place sequence he requires. 5. Best for Fixed Clamper Gripper 6. Harder to create 7. Best Sequence Control 8. Multiple Gripper Pickup Orientations 9. Multiple Gripper Pickup Groups</td>
<td>No</td>
<td>Manual</td>
<td>Manual</td>
<td>Any pick and place order desired</td>
</tr>
</tbody>
</table>

**Fig. 5-38: Auto Sequencer**
Fig. 5-39: Box Order Override
6 Advanced Programming (XML)

The EasyPallet® program is based on an XML core. Both the pendant application and pc-based control panel are used to update and write to these XML files. All box, pallet, and layer patterns are all stored within XML database files. EasyPallet® interacts with these files to produce pallet patterns and ultimately robot pallet jobs.

Advanced users familiar with XML programming, may find it easier to modify these XML files directly when making large changes. While it is very easy and intuitive to modify or add a box definition using either the EasyPallet® pendant application or pc-based control panel, it would be very time consuming to add a larger number of new box definitions in this manner. By modifying the BoxDB.xml file directly, the advanced user can make these changes very quickly.

The job templates used to structure the final palletizing jobs can also be modified to call other jobs or functions, add additional steps to clear process equipment, or any other additions required by your particular processes.

---

**CAUTION**

The operation of EasyPallet® is dependent upon the structure of these XML files. Any changes made to the XML configuration files and/or file structure will alter the operation of EasyPallet®. Motoman recommends you do not modify the original files and/or file configuration that came with your system. If modifications need to be made, they should be made to copies of these files and not to the originals. Modifications should only be performed by advanced users who are familiar with the operation of this software and XML programming.

If you have questions concerning advanced programming of EasyPallet®, please contact Motoman Customer Support.

6.1 Job Templates

EasyPallet® uses XML job templates to produce the set of palletizing jobs (Master, Pick, Place, etc.). They act as macros enabling similar operations to be performed at different locations (i.e. box pickup and placement sequences repeated at different locations).

These xml job templates are the core of EasyPallet® and set the format and structure of the resulting robot jobs. The xml job templates determine the robot job structure (joint or relative), and convert pallet data (box size, layer pattern, pallet pattern, robot type, gripper type, pallet location, pallet size, etc.) into the final set of robot jobs.
The EasyPallet® xml job templates are bundled as a set with each template file creating a separate robot .jbi job file. Each set of palletizing job templates contains the following individual xml job templates:

- Initialization Template - initializes the set of B-variables for a pallet station and pallet pattern.
- Increment Template - increments the set of B-variables after each pick and place operation.
- Master Template - coordinates job activity.
- Pick Template - performs box pickup operations.
- Place Template - performs box place operations.

These basic templates can easily be modified to provide a wide range of customization. In addition, new template sets can be created to meet a variety of different situations including:

- High speed placement
- Special approach points for a high conveyor/low pallet configuration
- Pickup sequence with check and re-pick if box is dropped at pickup
- I/O driven Master Job
- Master Job with empty pallet handling
- Master Job with custom checks for safety equipment
- Simple pickup job for use with no split drops

The EasyPallet® job templates are installed by default in the following directory:

`C:\Program Files\Motoman\EasyPalletPC\CustomTemplates\MotoPalletBased`

- **LookAhead.xml** - creates a job that tells the conveyor how many boxes to release on the next Pickup operation.
- **LookAheadInit.xml** - creates a job that tells the conveyor how many boxes to release for the first pickup operation of an new pallet operation.
- **SlipUp.xml** - creates jobs for slipsheet pickup operations
- **INITCAL** - add appropriate B and D Variables together so that the correct Initialization job is called
- **PickCal** - add appropriate B and D Variables together so that the correct Pickup jobs is called
- **PlaceCal** - appropriate B and D Variables together so that the correct Pickup job is called
- **PIPOVR** - allows users to hard code B Variables which will set the determine Where to pick and place the boxes and in which pattern number to use.

**NOTE**

PIPOVR is typically used during initial testing or when a PLC is not being used to control the system.

**PIPLSET** - reads IO and places these values into the appropriate B Variables. These B Variables determine Where to pick and place the boxes and in which pattern number to use.
The master job uses 5 standard jobs to call any job for any pallet in a cell. The specific pallet pattern jobs are all named using numbers only. The master job uses a set of helper jobs, and a set of variables to calculate the correct pick, place and initialization jobs to call.

The following flow chart shows the basic operation of the master job. As shown, the master job waits for Pick/Place command from the PLC. When a command is received, EasyPallet® checks if this is a new pallet. If it is a new pallet, the INITCAL job converts the commands to the appropriate number and calls the initialization job (via the number).

Next Pick and Place jobs are call, again by number.
Example:
Pallet # = 1
Conveyor # = 2
Pattern # = 22

Simplified Master Job

Master Job

Call P/P:PLSET or P/P:PLS

Valid Cmd?

No

New Pallet?

Yes

Call INITCAL

1. Call Job to initialize the pallet
2. Example 14000022.jbl

New Pallet?

Yes

Call P/P:PLSET or P/P:PLS

1. Call Pickup Job
2. Creates D Variable = 11200022
3. Call 112000022

New Pallet?

Yes

Stop
6.1.1 Job Numbering Scheme

All pallet pattern or plugin jobs adhere to the following numbering scheme. This number scheme is used to ensure that multiple pallet patterns can reside in the robot controller simultaneously.

XYZ00AAA.jbi

Where:

• X is the Pallet Number -- (between 1 and 8)
• Y is the job type
• Z meaning changes with Y -- See table below for specific definitions
• AAA is pallet pattern number from 1 to 999

<table>
<thead>
<tr>
<th>Y</th>
<th>Job Type</th>
<th>Z</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Pick Job</td>
<td>1 - 8</td>
<td>Conveyor Station #</td>
</tr>
<tr>
<td>2</td>
<td>Place Job</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Increment State Variables Job</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Initiate State Variables Job</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Information Summary Job</td>
<td>Creates a job with user specified summary Information</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Reserved</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Slip Pickup Jobs</td>
<td>1 - 8</td>
<td>Slip Sheet Station #</td>
</tr>
<tr>
<td>8</td>
<td>Slip Place Jobs</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Look Ahead Jobs</td>
<td>0</td>
<td>Contains all Look Ahead Values except the first one</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Look Ahead for first pickup of the first layer</td>
</tr>
</tbody>
</table>

Examples:

"11100101.jbi - Pallet 1, Pickup Job for Conveyor #1, job pattern # 101"
"3400122.jbi -- Pallet 3, Initialize State Variables job for pattern # 122"
"1200088.jbi -- Pallet 1, place job for pattern # 88"
"57100473.jbi - Pallet 5, Slip Pickup Job for pattern # 473"
"29100722.jbi - Pallet 2, Look Ahead Initialization job for pattern 722"
6.1.2 XML Template to Robot Job Conversion Process

The xml job templates and pallet data are converted into robot jobs in two separate steps. First, the template files are merged with the position data for each pick and place sequence required for the robot to build the pallet. During this conversion, Meta Commands ($ Commands), acting as placeholders within the xml job templates, are replaced with real position data or values.

Next, the XML files are converted into robot jobs. This process is similar to that used by Motoman's Points ImporterEG converter software to convert XML files into robot jobs.

6.1.3 Meta Commands and Variables

Merging the template files with the position data is accomplished using a macro operation with Meta commands and variables. The position data and template values are associated with variable names. The variable values are then retrieved in the template files by referencing the variable name with a dollar sign prefixed. The Meta commands replace the placeholder information with actual data or position information during the job creation process.

Meta commands and variables are prefixed by a dollar symbol ($) in the template files. EasyPallet® currently supports the following Meta commands:

<table>
<thead>
<tr>
<th>Meta Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Variable</td>
<td>Replaces the variable with the current value.</td>
</tr>
<tr>
<td>$*</td>
<td>Multiplies two $Variables together</td>
</tr>
<tr>
<td>$(and $)</td>
<td>Grouping Meta Operations</td>
</tr>
<tr>
<td>$+</td>
<td>Adds two $Variables together</td>
</tr>
</tbody>
</table>
6.1.3.1 Meta Command and Variable Examples

Example 1

In this example, the value of $x$ is added to the product of $\text{VectorXMagnitude}$, $\text{VectorX}$, and -1.

\[
X = 275.5 \\
\text{VectorXMagnitude} = 75 \\
\text{VectorX} = 1 \text{ (possible values are } 1, 0, -1) \\
\]

\[
<X> \$X \$+ \$\text{VectorXMagnitude} \$\times \$\text{VectorX} \$\times \$-1 \$ </X>
\]

\[
<X> 275.5 + (75 \times 1 \times -1) </X>
\]

This yields an instruction at runtime of: \(<X>200.5</X>\)

Example 2

In this example, the value of $x$ is added to the product of $\text{VectorXMagnitude}$, $\text{VectorX}$, and -1.

\[
Z = 634.5 \\
\text{BoxHeight} = 150 \\
\text{CurrentLayerCounter} = 3 \\
\text{SafeClearance} = 25 \\
\]

\[
<Z> \$Z \$+ \$\text{BoxHeight} \$\times \$\text{CurrentLayerCounter} \$ \$\text{SafeClearance}</Z>
\]

\[
<Z> 634.5 + (150 \times 3) + 25 </Z>
\]

This yields an instruction at runtime of: \(<Z>1109.5</Z>\)

6.1.3.2 Processing Instructions

Template files also have a small number of header type or general instructions. These instructions and their function are shown below:

<table>
<thead>
<tr>
<th>ProcessingInstructionTemplateName</th>
<th>Should always be $\text{MotoPallet}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProcessingInstructionJobType</td>
<td>This instruction gives the job type: Master, Pick, Place, Init, Inc</td>
</tr>
</tbody>
</table>
| ProcessingInstruction             | Pickup Jobs Options $\text{PickSinglePosition}$ - Use for simple jobs. Produces a pick job with only one pickup sequence for the complete pallet. Not appropriate for patterns requiring Asynchronous Placements.
|                                  | $\text{PickLayerPosition}$ - creates a pickup job for each layer.
|                                  | $\text{PickCustomPosition}$ - Creates a hard coded pick location for each pickup in the pallet. |
| ProcessingInstruction             | $\text{Repeat}, \text{EndRepeat}$ |
| ProcessingInstructionIf           | Tests the case. If the case is true, it executes all the statements until it finds the EndIf instruction |
| ProcessingInstructionEndIf        | Signifies the end of the if statement |
6.1.4 Editing a Template File

To edit a template file, proceed as follows:

1. Make a copy of the directory.
2. Open the desired file with XML NotePad Editor or any other XML editing software.
3. Make the changes you desire and save the file.
4. Test your changes.

6.1.4.1 Example 1: Modifying the Pickup Sequence

Problem:

The pickup conveyor is higher than the pallet and the pallet is positioned close to the conveyor. When executing the first layer place sequences, the robot gripper may collide with the conveyor when returning from the pallet to the conveyor. See Figure below:

Solution:

Add a lateral move to the start and end of the pickup sequence to move the robot and gripper safely away clear of the conveyor.

NOTE: The pickup sequence is in conveyor coordinates so that a lateral move towards the pallet would be in the positive Y and the negative X directions.
Steps:

1. Make a copy of the template directory.
2. Open the Pickup template in XML Notepad editor.
3. Copy the first position command.
4. Paste the position command back into the document. At this point you have two commands at the same location.
   a) Add a 300 mm offset to the Y Coordinate -> $+ $300
   b) Add a -300 mm offset to the X Coordinate -> $+ $-300
5. Find, copy and the last position command in the Pickup template.
6. Paste the position command back into the document. At this point you have two commands at the same location.
   a) Add a 300 mm offset to the last position command, Y Coordinate -> $+ $300
   b) Add a -300 mm offset to the last position command, X Coordinate -> $+ $-300
7. Save the modified Pickup template
8. Create new pallet jobs using the new template.
9. Carefully test the new jobs verifying that the new gripper motion safely clears the conveyor.

Fig. 6-40: Robot at End of Modified Pickup Sequence
6.1.4.2 Example 2: Adding a Call to Load Empty Pallet

**Problem:**
The current pallet job does not load an empty pallet when the full pallet is removed from the workcell.

**Solution:**
Add a call to a job that moves an empty pallet to the correct location in the palletizing cell when a full pallet is removed from the workcell.

**Steps:**
1. Make a copy of the template directory.
2. Open the Master template in the XML Notepad editor.
3. Add the following commands after the `<GenInform>` command in the Master template:

   `<GenInform>\JUMP *MAIN IF B009<![CDATA[<>1]]></GenInform>
   Line A <CallJob>LoadPall</CallJob>
   Line B <GenInform>\JUMP *MAIN</GenInform>

   **Line A**
   Calls the LoadPall robot job.

   **Line B**
   Jumps the program back to the *Main label.

4. Save the modified Master template.
6.1.5 MasterTemp.XML

```
<Root>
  <ProcessingInstructionTemplateType>$MotoPallet</ProcessingInstructionTemplateType>
  <ProcessingInstructionJobType>$Master</ProcessingInstructionJobType>
  <ProcessingInstructionFileName>MASTERX</ProcessingInstructionFileName>
  <GenInform>'Created from Master.xml'</GenInform>
  <GenInform>SET $PlaceOnPalletVar $PlaceOnPalletCmd</GenInform>
  <GenInform>SET $PickFromConveyerVar $PickFromConveyorCmd</GenInform>
  <GenInform>SET $PickPlaceDurationVar 1</GenInform>
  <CallJob>$CallType_INIT</CallJob>
  <GenInform>*MAIN</GenInform>
  <CallJob>$CallType_PICK</CallJob>
  <CallJob>$CallType_PLACE</CallJob>
  <GenInform>JUMP *MAIN IF B009<![[CDATA[<>1]]></GenInform>
</Root>
```

6.1.6 Velocity Control with D Variables

EasyPallet® allows you to use D-variables to change the robot's velocity at runtime without needing to remake the jobs. For example, you can lower the velocity during the first run to minimize the possibility of a robot crash during cell setup.

To setup your jobs to use D-variable velocities, proceed as follows:

1. From the Template directory, open the desired template file.
2. Locate the Velocity tags and replace the velocity number with a D variable as shown below.
6 Advanced Programming (XML)

6.2 Box Definition Database

Box definitions are stored in the BoxDB.xml file. Boxes can be modified and updated directly in this file. Additional boxes can be added simply by cut and paste and modifying the box name and dimensions.
6 Advanced Programming (XML)
6.3 Gripper Definitions Database

6.2.1 BoxDB.XML

<?xml version="1.0" standalone="yes"?>
<Root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\DOCUME~1\GREGW~1\MYDOCU~1\VISUAL~1\Projects\PointsImporter2005DevelopmentBaseForEasyPallet\GCodeRelease2version2003\bin\Dir_AllXMLSchemas\BoxDBSchemaV1.xsd">
  <Box>
    <BoxName>Example1</BoxName>
    <BoxLength>310</BoxLength>
    <BoxWidth>305</BoxWidth>
    <BoxHeight>165</BoxHeight>
  </Box>
  <Box>
    <BoxName>Example2</BoxName>
    <BoxLength>250</BoxLength>
    <BoxWidth>150</BoxWidth>
    <BoxHeight>125</BoxHeight>
  </Box>
  <Box>
    <BoxName>Example3</BoxName>
    <BoxLength>200</BoxLength>
    <BoxWidth>150</BoxWidth>
    <BoxHeight>125</BoxHeight>
  </Box>
  <Box>
    <BoxName>Example4</BoxName>
    <BoxLength>150</BoxLength>
    <BoxWidth>100</BoxWidth>
    <BoxHeight>75</BoxHeight>
  </Box>
  <Box>
    <BoxName>Example5</BoxName>
    <BoxLength>300</BoxLength>
    <BoxWidth>200</BoxWidth>
    <BoxHeight>100</BoxHeight>
  </Box>
</Root>

6.3 Gripper Definitions Database

Gripper definitions are stored in the GripperDB.xml file. Grippers can be modified and updated directly in this file. Additional grippers can be added simply by cut and paste and modifying the gripper name and definition.

6.3.1 GripperDB.XML

<?xml version="1.0" encoding="UTF-8"?>
<-- edited with XMLSpy v2007 sp2 (http://www.altova.com) by jeff magnuson (Motoman Inc.) -->
<-- edited with XMLSpy v2004 rel. 3 U (http://www.xmlspy.com) by Greg Webb NMN Webb (Motoman Inc.) -->
<Root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\PROGRA~1\Motoman\EasyPallet\PC2\Dir_AllXMLSchemas\GripperSchemaV1.xsd">
  <Gripper>
    <GripperName>Vac1BoxStoS</GripperName>
    <GripperType>Suction</GripperType>
    <MaxBoxes>1</MaxBoxes>
    <Layout>2</Layout>
</Root>
<BoxOrientation>SidetoSide</BoxOrientation>
<GripperSizeX>450</GripperSizeX>
<GripperSizeY>200</GripperSizeY>
<TCpx>0</TCpx>
<TCpy>0</TCpy>
<TCpz>200</TCpz>
<TCptx>180</TCptx>
<TCpty>0</TCpty>
<TCptz>0</TCptz>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>1</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>Vac1BoxEtoE</GripperName>
<GripperType>Suction</GripperType>
<MaxBoxes>1</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>EndtoEnd</BoxOrientation>
<GripperSizeX>600</GripperSizeX>
<GripperSizeY>300</GripperSizeY>
<TCpx>0</TCpx>
<TCpy>0</TCpy>
<TCpz>200</TCpz>
<TCptx>180</TCptx>
<TCpty>0</TCpty>
<TCptz>0</TCptz>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>1</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>Vac2BoxStoS</GripperName>
<GripperType>Suction</GripperType>
<MaxBoxes>2</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>SidetoSide</BoxOrientation>
<GripperSizeX>450</GripperSizeX>
<GripperSizeY>200</GripperSizeY>
<TCpx>0</TCpx>
<TCpy>0</TCpy>
<TCpz>200</TCpz>
<TCptx>180</TCptx>
<TCpty>0</TCpty>
<TCptz>0</TCptz>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>2</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>Vac3BoxStoS</GripperName>
<GripperType>Suction</GripperType>
<MaxBoxes>3</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>SidetoSide</BoxOrientation>
<GripperSizeX>500</GripperSizeX>
<GripperSizeY>200</GripperSizeY>
<TCpx>0</TCpx>
<TCpy>0</TCpy>
<TCpz>200</TCpz>
<TCptx>180</TCptx>

6 Advanced Programming (XML)
6.3 Gripper Definitions Database
<Gripper>
  <GripperName>Vac4BoxStoS</GripperName>
  <GripperType>Suction</GripperType>
  <MaxBoxes>4</MaxBoxes>
  <Layout>2</Layout>
  <BoxOrientation>SidetoSide</BoxOrientation>
  <GripperSizeX>450</GripperSizeX>
  <GripperSizeY>200</GripperSizeY>
  <TCPX>0</TCPX>
  <TCPY>0</TCPY>
  <TCPZ>200</TCPZ>
  <TCPtX>180</TCPtX>
  <TCPtY>0</TCPtY>
  <TCPtZ>0</TCPtZ>
  <OffsetRotation>0</OffsetRotation>
  <XQuantBoxes>3</XQuantBoxes>
  <YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
  <GripperName>Vac2BoxEtoE</GripperName>
  <GripperType>Suction</GripperType>
  <MaxBoxes>2</MaxBoxes>
  <Layout>2</Layout>
  <BoxOrientation>EndtoEnd</BoxOrientation>
  <GripperSizeX>600</GripperSizeX>
  <GripperSizeY>300</GripperSizeY>
  <TCPX>0</TCPX>
  <TCPY>0</TCPY>
  <TCPZ>200</TCPZ>
  <TCPtX>180</TCPtX>
  <TCPtY>0</TCPtY>
  <TCPtZ>0</TCPtZ>
  <OffsetRotation>0</OffsetRotation>
  <XQuantBoxes>2</XQuantBoxes>
  <YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
  <GripperName>Vac3BoxEtoE</GripperName>
  <GripperType>Suction</GripperType>
  <MaxBoxes>3</MaxBoxes>
  <Layout>2</Layout>
  <BoxOrientation>EndtoEnd</BoxOrientation>
  <GripperSizeX>600</GripperSizeX>
  <GripperSizeY>300</GripperSizeY>
  <TCPX>0</TCPX>
  <TCPY>0</TCPY>
  <TCPZ>200</TCPZ>
  <TCPtX>180</TCPtX>
  <TCPtY>0</TCPtY>
  <TCPtZ>0</TCPtZ>
  <OffsetRotation>0</OffsetRotation>
  <XQuantBoxes>3</XQuantBoxes>
  <YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>Vac4BoxEtoE</GripperName>
<GripperType>Suction</GripperType>
<MaxBoxes>4</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>EndtoEnd</BoxOrientation>
<GripperSizeX>600</GripperSizeX>
<GripperSizeY>200</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCPtX>180</TCPtX>
<TCPtY>0</TCPtY>
<TCPtZ>0</TCPtZ>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>4</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>Vac4BoxSquareEtoE</GripperName>
<GripperType>Suction</GripperType>
<MaxBoxes>4</MaxBoxes>
<Layout>1</Layout>
<BoxOrientation>EndtoEnd</BoxOrientation>
<GripperSizeX>600</GripperSizeX>
<GripperSizeY>200</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCPtX>180</TCPtX>
<TCPtY>0</TCPtY>
<TCPtZ>0</TCPtZ>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>2</XQuantBoxes>
<YQuantBoxes>2</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>Vac4BoxSquareStoS</GripperName>
<GripperType>Suction</GripperType>
<MaxBoxes>4</MaxBoxes>
<Layout>1</Layout>
<BoxOrientation>SidetoSide</BoxOrientation>
<GripperSizeX>600</GripperSizeX>
<GripperSizeY>200</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCPtX>180</TCPtX>
<TCPtY>0</TCPtY>
<TCPtZ>0</TCPtZ>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>2</XQuantBoxes>
<YQuantBoxes>2</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>Clamshell1BoxStoS</GripperName>
<GripperType>ClamShell</GripperType>
<MaxBoxes>1</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>SidetoSide</BoxOrientation>
<GripperSizeX>600</GripperSizeX>
<GripperSizeY>200</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCptX>180</TCptX>
<TCptY>0</TCptY>
<TCptZ>0</TCptZ>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>1</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>Clamshell2BoxStoS</GripperName>
<GripperType>ClamShell</GripperType>
<MaxBoxes>2</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>SidetoSide</BoxOrientation>
<GripperSizeX>450</GripperSizeX>
<GripperSizeY>200</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCptX>180</TCptX>
<TCptY>0</TCptY>
<TCptZ>0</TCptZ>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>2</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>Clamshell3BoxStoS</GripperName>
<GripperType>ClamShell</GripperType>
<MaxBoxes>3</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>SidetoSide</BoxOrientation>
<GripperSizeX>500</GripperSizeX>
<GripperSizeY>200</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCptX>180</TCptX>
<TCptY>0</TCptY>
<TCptZ>0</TCptZ>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>3</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>Clamshell4BoxStoS</GripperName>
<GripperType>ClamShell</GripperType>
<MaxBoxes>4</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>SidetoSide</BoxOrientation>
<GripperSizeX>450</GripperSizeX>
<GripperSizeY>200</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCptX>180</TCptX>
<TCptY>0</TCptY>
<TCptZ>0</TCptZ>
<OffsetRotation>0</OffsetRotation>
<Gripper>
  <GripperName>Clamshell1BoxEtoE</GripperName>
  <GripperType>ClamShell</GripperType>
  <MaxBoxes>1</MaxBoxes>
  <Layout>2</Layout>
  <BoxOrientation>EndtoEnd</BoxOrientation>
  <GripperSizeX>600</GripperSizeX>
  <GripperSizeY>300</GripperSizeY>
  <TCPX>0</TCPX>
  <TCPY>0</TCPY>
  <TCPZ>200</TCPZ>
  <TCPtX>180</TCPtX>
  <TCPtY>0</TCPtY>
  <TCPtZ>0</TCPtZ>
  <OffsetRotation>0</OffsetRotation>
</Gripper>

<Gripper>
  <GripperName>Clamshell2BoxEtoE</GripperName>
  <GripperType>ClamShell</GripperType>
  <MaxBoxes>2</MaxBoxes>
  <Layout>2</Layout>
  <BoxOrientation>EndtoEnd</BoxOrientation>
  <GripperSizeX>600</GripperSizeX>
  <GripperSizeY>300</GripperSizeY>
  <TCPX>0</TCPX>
  <TCPY>0</TCPY>
  <TCPZ>200</TCPZ>
  <TCPtX>180</TCPtX>
  <TCPtY>0</TCPtY>
  <TCPtZ>0</TCPtZ>
  <OffsetRotation>0</OffsetRotation>
</Gripper>

<Gripper>
  <GripperName>Clamshell3BoxEtoE</GripperName>
  <GripperType>ClamShell</GripperType>
  <MaxBoxes>3</MaxBoxes>
  <Layout>2</Layout>
  <BoxOrientation>EndtoEnd</BoxOrientation>
  <GripperSizeX>600</GripperSizeX>
  <GripperSizeY>300</GripperSizeY>
  <TCPX>0</TCPX>
  <TCPY>0</TCPY>
  <TCPZ>200</TCPZ>
  <TCPtX>180</TCPtX>
  <TCPtY>0</TCPtY>
  <TCPtZ>0</TCPtZ>
  <OffsetRotation>0</OffsetRotation>
</Gripper>

<Gripper>
  <GripperName>Clamshell4BoxEtoE</GripperName>
  <GripperType>ClamShell</GripperType>
  <MaxBoxes>4</MaxBoxes>
</Gripper>
<Gripper>
  <GripperName>SymetricFork1BoxStoS</GripperName>
  <GripperType>SymetricFork</GripperType>
  <MaxBoxes>1</MaxBoxes>
  <Layout>2</Layout>
  <BoxOrientation>SidetoSide</BoxOrientation>
  <GripperSizeX>450</GripperSizeX>
  <GripperSizeY>200</GripperSizeY>
  <TCPX>0</TCPX>
  <TCPY>0</TCPY>
  <TCPZ>200</TCPZ>
  <TCPtX>180</TCPtX>
  <TCPtY>0</TCPtY>
  <TCPtZ>0</TCPtZ>
  <OffsetRotation>0</OffsetRotation>
  <XQuantBoxes>4</XQuantBoxes>
  <YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
  <GripperName>SymetricFork2BoxStoS</GripperName>
  <GripperType>SymetricFork</GripperType>
  <MaxBoxes>2</MaxBoxes>
  <Layout>2</Layout>
  <BoxOrientation>SidetoSide</BoxOrientation>
  <GripperSizeX>450</GripperSizeX>
  <GripperSizeY>200</GripperSizeY>
  <TCPX>0</TCPX>
  <TCPY>0</TCPY>
  <TCPZ>200</TCPZ>
  <TCPtX>180</TCPtX>
  <TCPtY>0</TCPtY>
  <TCPtZ>0</TCPtZ>
  <OffsetRotation>0</OffsetRotation>
  <XQuantBoxes>2</XQuantBoxes>
  <YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
  <GripperName>SymetricFork3BoxStoS</GripperName>
  <GripperType>SymetricFork</GripperType>
  <MaxBoxes>3</MaxBoxes>
  <Layout>2</Layout>
  <BoxOrientation>SidetoSide</BoxOrientation>
  <GripperSizeX>500</GripperSizeX>
  <GripperSizeY>200</GripperSizeY>
  <TCPX>0</TCPX>
  <TCPY>0</TCPY>
  <TCPZ>200</TCPZ>
</Gripper>
<TCPtX>180</TCPtX>
<TCPtY>0</TCPtY>
<TCPtZ>0</TCPtZ>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>3</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>SymmetricFork4BoxStoS</GripperName>
<GripperType>SymmetricFork</GripperType>
<MaxBoxes>4</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>SideToSide</BoxOrientation>
<GripperSizeX>450</GripperSizeX>
<GripperSizeY>200</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCPtX>180</TCPtX>
<TCPtY>0</TCPtY>
<TCPtZ>0</TCPtZ>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>4</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>SymmetricFork1BoxEtoE</GripperName>
<GripperType>SymmetricFork</GripperType>
<MaxBoxes>1</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>EndToEnd</BoxOrientation>
<GripperSizeX>600</GripperSizeX>
<GripperSizeY>300</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCPtX>180</TCPtX>
<TCPtY>0</TCPtY>
<TCPtZ>0</TCPtZ>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>1</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>SymmetricFork2BoxEtoE</GripperName>
<GripperType>SymmetricFork</GripperType>
<MaxBoxes>2</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>EndToEnd</BoxOrientation>
<GripperSizeX>600</GripperSizeX>
<GripperSizeY>300</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCPtX>180</TCPtX>
<TCPtY>0</TCPtY>
<TCPtZ>0</TCPtZ>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>2</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>
<Gripper>
  <GripperName>SymetricFork3BoxEtoE</GripperName>
  <GripperType>SymetricFork</GripperType>
  <MaxBoxes>3</MaxBoxes>
  <Layout>2</Layout>
  <BoxOrientation>EndtoEnd</BoxOrientation>
  <GripperSizeX>600</GripperSizeX>
  <GripperSizeY>300</GripperSizeY>
  <TCPX>0</TCPX>
  <TCPY>0</TCPY>
  <TCPZ>200</TCPZ>
  <TCPtX>180</TCPtX>
  <TCPtY>0</TCPtY>
  <TCPtZ>0</TCPtZ>
  <OffsetRotation>0</OffsetRotation>
  <XQuantBoxes>3</XQuantBoxes>
  <YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
  <GripperName>SymetricFork4BoxEtoE</GripperName>
  <GripperType>SymetricFork</GripperType>
  <MaxBoxes>4</MaxBoxes>
  <Layout>2</Layout>
  <BoxOrientation>EndtoEnd</BoxOrientation>
  <GripperSizeX>600</GripperSizeX>
  <GripperSizeY>200</GripperSizeY>
  <TCPX>0</TCPX>
  <TCPY>0</TCPY>
  <TCPZ>200</TCPZ>
  <TCPtX>180</TCPtX>
  <TCPtY>0</TCPtY>
  <TCPtZ>0</TCPtZ>
  <OffsetRotation>0</OffsetRotation>
  <XQuantBoxes>4</XQuantBoxes>
  <YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
  <GripperName>FixedFork1BoxStoS</GripperName>
  <GripperType>FixedFork</GripperType>
  <MaxBoxes>1</MaxBoxes>
  <Layout>2</Layout>
  <BoxOrientation>SidetoSide</BoxOrientation>
  <GripperSizeX>450</GripperSizeX>
  <GripperSizeY>200</GripperSizeY>
  <TCPX>0</TCPX>
  <TCPY>0</TCPY>
  <TCPZ>200</TCPZ>
  <TCPtX>180</TCPtX>
  <TCPtY>0</TCPtY>
  <TCPtZ>0</TCPtZ>
  <OffsetRotation>0</OffsetRotation>
  <XQuantBoxes>1</XQuantBoxes>
  <YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
  <GripperName>FixedFork2BoxStoS</GripperName>
  <GripperType>FixedFork</GripperType>
  <MaxBoxes>2</MaxBoxes>
  <Layout>2</Layout>
  <BoxOrientation>SidetoSide</BoxOrientation>
  <GripperSizeX>450</GripperSizeX>
<GripperSizeY>200</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCPtX>180</TCPtX>
<TCPtY>0</TCPtY>
<TCPtZ>0</TCPtZ>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>2</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>FixedFork3BoxStoS</GripperName>
<GripperType>FixedFork</GripperType>
<MaxBoxes>3</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>SidetoSide</BoxOrientation>
<GripperSizeX>500</GripperSizeX>
<GripperSizeY>200</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCPtX>180</TCPtX>
<TCPtY>0</TCPtY>
<TCPtZ>0</TCPtZ>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>3</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>FixedFork4BoxStoS</GripperName>
<GripperType>FixedFork</GripperType>
<MaxBoxes>4</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>SidetoSide</BoxOrientation>
<GripperSizeX>450</GripperSizeX>
<GripperSizeY>200</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCPtX>180</TCPtX>
<TCPtY>0</TCPtY>
<TCPtZ>0</TCPtZ>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>4</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>FixedFork1BoxEtoE</GripperName>
<GripperType>FixedFork</GripperType>
<MaxBoxes>1</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>EndtoEnd</BoxOrientation>
<GripperSizeX>600</GripperSizeX>
<GripperSizeY>300</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCPtX>180</TCPtX>
<TCPtY>0</TCPtY>
<TCPtZ>0</TCPtZ>
<Gripper>
<GripperName>FixedFork2BoxEtoE</GripperName>
<GripperType>FixedFork</GripperType>
<MaxBoxes>2</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>EndtoEnd</BoxOrientation>
<GripperSizeX>600</GripperSizeX>
<GripperSizeY>300</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCPtX>180</TCPtX>
<TCPtY>0</TCPtY>
<TCPtZ>0</TCPtZ>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>1</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>FixedFork3BoxEtoE</GripperName>
<GripperType>FixedFork</GripperType>
<MaxBoxes>3</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>EndtoEnd</BoxOrientation>
<GripperSizeX>600</GripperSizeX>
<GripperSizeY>300</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCPtX>180</TCPtX>
<TCPtY>0</TCPtY>
<TCPtZ>0</TCPtZ>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>2</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>FixedFork4BoxEtoE</GripperName>
<GripperType>FixedFork</GripperType>
<MaxBoxes>1</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>SidetoSide</BoxOrientation>
<GripperSizeX>450</GripperSizeX>
<GripperSizeY>200</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCPtX>180</TCPtX>
<TCPtY>0</TCPtY>
<TCPtZ>0</TCPtZ>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>1</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>Shovel1BoxStoS</GripperName>
<GripperType>Shovel</GripperType>
<MaxBoxes>4</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>EndtoEnd</BoxOrientation>
<GripperSizeX>600</GripperSizeX>
<GripperSizeY>200</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCPtX>180</TCPtX>
<TCPtY>0</TCPtY>
<TCPtZ>0</TCPtZ>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>4</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>Shovel2BoxStoS</GripperName>
<GripperType>Shovel</GripperType>
<MaxBoxes>2</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>SidetoSide</BoxOrientation>
<GripperSizeX>450</GripperSizeX>
<GripperSizeY>200</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCPtX>180</TCPtX>
<TCPtY>0</TCPtY>
<TCPtZ>0</TCPtZ>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>2</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>Shovel3BoxStoS</GripperName>
<GripperType>Shovel</GripperType>
<MaxBoxes>3</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>SidetoSide</BoxOrientation>
<GripperSizeX>500</GripperSizeX>
<GripperSizeY>200</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<TCPZ>200</TCPZ>
<TCPtX>180</TCPtX>
<TCPtY>0</TCPtY>
<TCPtZ>0</TCPtZ>
<OffsetRotation>0</OffsetRotation>
<XQuantBoxes>3</XQuantBoxes>
<YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
<GripperName>Shovel4BoxStoS</GripperName>
<GripperType>Shovel</GripperType>
<MaxBoxes>4</MaxBoxes>
<Layout>2</Layout>
<BoxOrientation>SidetoSide</BoxOrientation>
<GripperSizeX>450</GripperSizeX>
<GripperSizeY>200</GripperSizeY>
<TCPX>0</TCPX>
<TCPY>0</TCPY>
<Gripper>
  <GripperName>Shovel1BoxEndtoEnd</GripperName>
  <GripperType>Shovel</GripperType>
  <MaxBoxes>1</MaxBoxes>
  <Layout>2</Layout>
  <BoxOrientation>EndtoEnd</BoxOrientation>
  <GripperSizeX>600</GripperSizeX>
  <GripperSizeY>300</GripperSizeY>
  <TCPX>0</TCPX>
  <TCPY>0</TCPY>
  <TCPZ>200</TCPZ>
  <TCPtX>180</TCPtX>
  <TCPtY>0</TCPtY>
  <TCPtZ>0</TCPtZ>
  <OffsetRotation>0</OffsetRotation>
  <XQuantBoxes>4</XQuantBoxes>
  <YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
  <GripperName>Shovel2BoxEndtoEnd</GripperName>
  <GripperType>Shovel</GripperType>
  <MaxBoxes>2</MaxBoxes>
  <Layout>2</Layout>
  <BoxOrientation>EndtoEnd</BoxOrientation>
  <GripperSizeX>600</GripperSizeX>
  <GripperSizeY>300</GripperSizeY>
  <TCPX>0</TCPX>
  <TCPY>0</TCPY>
  <TCPZ>200</TCPZ>
  <TCPtX>180</TCPtX>
  <TCPtY>0</TCPtY>
  <TCPtZ>0</TCPtZ>
  <OffsetRotation>0</OffsetRotation>
  <XQuantBoxes>2</XQuantBoxes>
  <YQuantBoxes>1</YQuantBoxes>
</Gripper>

<Gripper>
  <GripperName>Shovel3BoxEndtoEnd</GripperName>
  <GripperType>Shovel</GripperType>
  <MaxBoxes>3</MaxBoxes>
  <Layout>2</Layout>
  <BoxOrientation>EndtoEnd</BoxOrientation>
  <GripperSizeX>600</GripperSizeX>
  <GripperSizeY>300</GripperSizeY>
  <TCPX>0</TCPX>
  <TCPY>0</TCPY>
  <TCPZ>200</TCPZ>
  <TCPtX>180</TCPtX>
  <TCPtY>0</TCPtY>
  <TCPtZ>0</TCPtZ>
  <OffsetRotation>0</OffsetRotation>
  <XQuantBoxes>3</XQuantBoxes>
  <YQuantBoxes>1</YQuantBoxes>
</Gripper>
Pallet definitions are stored in the PalletDB.xml file. Pallets can be modified and updated directly in this file. Additional pallets can be added simply by cut and paste and modifying the pallet name and dimensions.

6.4.1 PalletDB.XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<Root>
    <Pallet>
        <PalletName>24x18x3_PE_2WAY</PalletName>
        <PalletLength>610</PalletLength>
        <PalletWidth>457</PalletWidth>
        <PalletHeight>75</PalletHeight>
        <Discription>Convoy I, 2-way forklift entry, 2-way hand truck entry</Discription>
        <Manufacturer>Menasha Corporation</Manufacturer>
        <Material>Polyethylene</Material>
        <Weight>6</Weight>
        <Load_00020_Height>110</Load_00020_Height>
        <Static_00020_Load>5000</Static_00020_Load>
        <Dynamic_00020_Load>3000</Dynamic_00020_Load>
        <Max_00020_Overhang>0</Max_00020_Overhang>
    </Pallet>
    <Pallet>
        <PalletName>30x24x6_PE_2WAY</PalletName>
        <PalletLength>762</PalletLength>
        <PalletWidth>610</PalletWidth>
        <PalletHeight>152</PalletHeight>
        <Discription>Convoy I, 2-way forklift entry, 2-way hand truck entry</Discription>
        <Manufacturer>Menasha Corporation</Manufacturer>
        <Material>Polyethylene</Material>
        <Weight>6</Weight>
        <Load_00020_Height>110</Load_00020_Height>
    </Pallet>
</Root>
```
<Pallet>
<PalletName>36x30x5.5_PE_4WAY</PalletName>
<PalletLength>914</PalletLength>
<PalletWidth>762</PalletWidth>
<PalletHeight>140</PalletHeight>
<Discription>Convoy I, 4-way forklift entry, 4-way hand truck entry</Discription>
<Manufacturer>Menasha Corporation</Manufacturer>
<Material>Polyethylene</Material>
<Weight>10</Weight>
<Loadx0020_Height>110</Loadx0020_Height>
<Static_x0020_Load>5000</Static_x0020_Load>
<Dynamic_x0020_Load>3000</Dynamic_x0020_Load>
<Max_x0020_Overhang>0</Max_x0020_Overhang>
</Pallet>
<Pallet>
<PalletName>36x36x5.5_PE_4WAY</PalletName>
<PalletLength>914</PalletLength>
<PalletWidth>914</PalletWidth>
<PalletHeight>140</PalletHeight>
<Discription>Convoy I, 4-way forklift entry, 4-way hand truck entry</Discription>
<Manufacturer>Menasha Corporation</Manufacturer>
<Material>Polyethylene</Material>
<Weight>12</Weight>
<Loadx0020_Height>110</Loadx0020_Height>
<Static_x0020_Load>10000</Static_x0020_Load>
<Dynamic_x0020_Load>3500</Dynamic_x0020_Load>
<Max_x0020_Overhang>0</Max_x0020_Overhang>
</Pallet>
<Pallet>
<PalletName>36x36x5.75_HDPE_4WAY</PalletName>
<PalletLength>914</PalletLength>
<PalletWidth>914</PalletWidth>
<PalletHeight>146</PalletHeight>
<Discription>Convoy II, 4-way forklift entry, 2-way hand truck entry</Discription>
<Manufacturer>Menasha Corporation</Manufacturer>
<Material>HDPE</Material>
<Weight>32</Weight>
<Loadx0020_Height>110</Loadx0020_Height>
<Static_x0020_Load>20000</Static_x0020_Load>
<Dynamic_x0020_Load>3000</Dynamic_x0020_Load>
<Max_x0020_Overhang>0</Max_x0020_Overhang>
</Pallet>
<Pallet>
<PalletName>40x28x5.25_HDPE_4WAY</PalletName>
<PalletLength>1016</PalletLength>
<PalletWidth>711</PalletWidth>
<PalletHeight>133</PalletHeight>
<Discription>Convoy I, 4-way forklift entry, 2-way hand truck entry</Discription>
<Manufacturer>Menasha Corporation</Manufacturer>
<Material>HDPE</Material>
<Weight>19</Weight>
<Loadx0020_Height>110</Loadx0020_Height>
<Static_x0020_Load>10000</Static_x0020_Load>
<Dynamic_x0020_Load>15000</Dynamic_x0020_Load>
<Max_x0020_Overhang>0</Max_x0020_Overhang>
<Pallet>
<PalletName>40x36x4.25_STEEL_4WAY</PalletName>
PalletLength:1016
PalletWidth:914
PalletHeight:108
<Discription>Econo-Master, 4-way forklift entry</Discription>
<Manufacturer>Banner Metals, Inc.</Manufacturer>
<Material>Steel</Material>
<Weight>44</Weight>
<Load_x0020_Height>110</Load_x0020_Height>
<Static_x0020_Load>40000</Static_x0020_Load>
<Dynamic_x0020_Load>4000</Dynamic_x0020_Load>
<Max_x0020_Overhang>0</Max_x0020_Overhang>
</Pallet>

<Pallet>
PalletName>40x36x6_STEEL_APC</PalletName>
PalletLength:1016
PalletWidth:914
PalletHeight:152
<Discription>APC</Discription>
<Manufacturer>Post Office</Manufacturer>
<Material>Steel</Material>
<Weight>50</Weight>
<Load_x0020_Height>110</Load_x0020_Height>
<Static_x0020_Load>5000</Static_x0020_Load>
<Dynamic_x0020_Load>1000</Dynamic_x0020_Load>
<Max_x0020_Overhang>0</Max_x0020_Overhang>
</Pallet>

<Pallet>
PalletName>40x40x4.25_STEEL_4WAY</PalletName>
PalletLength:1016
PalletWidth:1016
PalletHeight:108
<Discription>Econo-Master, 4-way forklift entry</Discription>
<Manufacturer>Banner Metals, Inc.</Manufacturer>
<Material>Steel</Material>
<Weight>46</Weight>
<Load_x0020_Height>110</Load_x0020_Height>
<Static_x0020_Load>40000</Static_x0020_Load>
<Dynamic_x0020_Load>4000</Dynamic_x0020_Load>
<Max_x0020_Overhang>0</Max_x0020_Overhang>
</Pallet>

<Pallet>
PalletName>40x40x5.5_PE_4WAY</PalletName>
PalletLength:1016
PalletWidth:1016
PalletHeight:140
<Discription>Convoy I, 4-way forklift entry, 4-way hand truck entry</Discription>
<Manufacturer>Menasha Corporation</Manufacturer>
<Material>Polyethylene</Material>
<Weight>14</Weight>
<Load_x0020_Height>110</Load_x0020_Height>
<Static_x0020_Load>15000</Static_x0020_Load>
<Dynamic_x0020_Load>3000</Dynamic_x0020_Load>
<Max_x0020_Overhang>0</Max_x0020_Overhang>
</Pallet>
6 Advanced Programming (XML)
6.4 Pallet Definition Database

```xml
</Pallet>
<Pallet>
  <PalletName>40x40x5.5_WOOD_4WAY</PalletName>
  <PalletLength>1016</PalletLength>
  <PalletWidth>1016</PalletWidth>
  <PalletHeight>140</PalletHeight>
  <Discription>4-way forklift entry, 4-way hand truck entry</Discription>
  <Manufacturer>xyz Corporation</Manufacturer>
  <Material>Wood</Material>
  <Weight>50</Weight>
  <Load_x0020_Height>110</Load_x0020_Height>
  <Static_x0020_Load>5000</Static_x0020_Load>
  <Dynamic_x0020_Load>1000</Dynamic_x0020_Load>
  <Max_x0020_Overhang>0</Max_x0020_Overhang>
</Pallet>
<Pallet>
```

154652-1CD