MotoSoft™

MotoSim Points Importer EG User’s Manual

Part Number: 148610-1CD
Revision: 0
COMPLETE OUR ONLINE SURVEY

Motoman is committed to total customer satisfaction! Please give us your feedback on the technical manuals you received with your Motoman robotic solution.

To participate, go to the following website:

http://www.motoman.com/forms/techpubs.asp
Table of Contents

Chapter 1
Introduction ................................................................. 1
1.1 Conversion Process Overview ............................................. 2
1.2 About This Document ................................................... 3
1.3 Features ........................................................................ 4
   1.3.1 Supported File Formats ................................................. 7
1.4 System Requirements .................................................... 8
1.5 Installing MotoSim Points Importer EG ................................. 9
   1.5.1 Installing the Hardware Key ........................................... 10
1.6 First Time Use ............................................................. 10
1.7 Learning MotoSim Points Importer EG ................................. 11
   1.7.1 Using Help ................................................................ 11
   1.7.2 Motoman Technical Education Center (MTEC) .................. 11
   1.7.3 XML Resources ......................................................... 12
1.8 Reference to Other Documentation .................................... 12
1.9 Customer Service Information ......................................... 12

Chapter 2
Safety ........................................................................... 13
2.1 Introduction ................................................................... 13
2.2 Standard Conventions .................................................... 14
2.3 General Safeguarding Tips ................................................. 15
2.4 Mechanical Safety Devices .............................................. 15
2.5 Installation Safety .......................................................... 15
2.6 Programming Safety ....................................................... 16
2.7 Operation Safety ............................................................ 17
2.8 Maintenance Safety ........................................................ 17

Chapter 3
MotoSim Points Importer EG ............................................. 19
3.1 Understanding the MotoSim Points Importer EG Work Area ....... 19
3.2 Command Menus .......................................................... 20
3.3 Process Buttons ............................................................ 20
3.4 Status Windows ............................................................ 21
3.5 Transparency ............................................................... 21
3.6 Working with MotoSim Points Importer EG .......................... 22
   3.6.1 Creating a Job ............................................................ 22
   3.6.2 Batch Job Processing .................................................. 22
Chapter 4
Points to Robot Job Converter .........................................................23
  4.1 Understanding the Points to Robot Job Converter Work Area ............24
    4.1.1 Command Menus .............................................................24
    4.1.2 Configuration Files Toolbar .................................................25
      4.1.2.1 Cell Configuration File ..............................................25
      4.1.2.2 Job Configuration File ..............................................26
      4.1.2.3 Points Configuration File ...........................................27
      4.1.2.4 Clear Configuration Files ...........................................27
      4.1.2.5 Reload Configuration Files ...........................................27
    4.1.3 Converter Setup Toolbar ..................................................28
      4.1.3.1 Select and Convert Points File ......................................28
      4.1.3.2 Play Job .....................................................................28
    4.1.4 Edit Toolbar .................................................................28
      4.1.4.1 Points File .................................................................28
      4.1.4.2 XML Files ..................................................................29
      4.1.4.3 INF Files ....................................................................29
      4.1.4.4 JBI Files ....................................................................29
      4.1.4.5 View Error File .............................................................29
      4.1.4.6 Job Directory ...............................................................29
    4.1.5 Transparency .....................................................................30
      4.1.5.1 On Top Check Box ........................................................30
    4.1.6 Fault Logging .....................................................................30
  4.2 Opening Points to Robot Job Converter ........................................31
  4.3 Getting Started (Quick Start) .....................................................31
    4.3.1 MotoSim EG Cell Setup .......................................................32
    4.3.2 Cell Configuration Setup .....................................................36
    4.3.3 Job Configuration Setup ......................................................37
    4.3.4 Points Configuration Setup ..................................................38
    4.3.5 Point File Conversion ..........................................................38

Chapter 5
G-Code to Robot Job Converter .......................................................41
  5.1 Installing G-Code Converter .......................................................42
  5.2 First Time Use .........................................................................42
    5.2.1 Sample Files .....................................................................42
  5.3 Understanding the G-Code Converter Work Area ............................43
    5.3.1 Command Menus ...............................................................43
    5.3.2 Process Toolbar ...............................................................44
    5.3.3 Status Bar .......................................................................44
    5.3.4 Tabbed Panels .................................................................45
5.3.4.1 XML Config File Panel .......................... .45
5.3.4.2 NC File Panel ...................................... .46
5.3.4.3 NC to XML Instructions Panel .......................... .47
5.3.4.4 Robot XML Files Panel .................................. .48
5.3.4.5 Robot JBI Files Panel .................................. .49
5.3.4.6 JBI Log Panel ........................................ .49
5.3.4.7 Summary Panel ......................................... .50
5.3.4.8 Config. Control Panel ................................... .50
5.3.5 Current CNC File Status Bar .............................. .51
5.3.6 Current Config File Status Bar ............................. .51
5.3.7 Transparency ............................................ .51
  5.3.7.1 On Top Check Box ................................... .51
5.4 Opening G-Code to Robot Job Converter ....................... .51
5.5 XML Configuration File ..................................... .52
  5.5.1 File Configuration> .................................... .52
    5.5.1.1 RegExParseInt ..................................... .55
    5.5.1.2 ParseGroups ...................................... .56
    5.5.1.3 StartFile ......................................... .56
    5.5.1.4 ThrowAwayLines ................................... .56
    5.5.1.5 ThrowAwayLinesRegEx ................................. .56
    5.5.1.6 ParseGroupsThrowAwayLines ......................... .56
    5.5.1.7 InformFileFormats ................................... .56
    5.5.1.8 AngleSystem ....................................... .57
    5.5.1.9 MotionFunctions ..................................... .57
    5.5.1.10 FourToSixAxis ..................................... .57
    5.5.1.11 Units .............................................. .58
    5.5.1.12 ThreeDOF ......................................... .58
    5.5.1.13 ReferenceModel .................................... .58
    5.5.1.14 ArmConfig ......................................... .58
    5.5.1.15 AlignXwithPath .................................... .58
    5.5.1.16 CircleFixUp ....................................... .58
    5.5.1.17 CircleIJCenter ..................................... .59
    5.5.1.18 DefaultToolNumber .................................. .59
    5.5.1.19 DefaultNCPath ..................................... .59
    5.5.1.20 DCISimpleDrip .................................... .59
    5.5.1.21 DCISimpleDripDVar ................................ .59
    5.5.1.22 CannedVelocity .................................... .59
    5.5.1.23 ExternalAxes ...................................... .59
  5.5.2 <GTOINFORM> .......................................... .59
    5.5.2.1 <GCODE> ........................................... .60
    5.5.2.2 <INFORM> .......................................... .61
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.6 Getting Started (Quick Start)</td>
<td>64</td>
</tr>
<tr>
<td>5.6.1 MotoSim EG Cell Setup</td>
<td>65</td>
</tr>
<tr>
<td>5.6.2 XML Configuration File Setup</td>
<td>69</td>
</tr>
<tr>
<td>5.6.3 Standard CNC File Conversion</td>
<td>75</td>
</tr>
<tr>
<td>5.6.4 Circles in CNC and Robots</td>
<td>75</td>
</tr>
<tr>
<td>5.6.4.1 Circles in CNCs</td>
<td>75</td>
</tr>
<tr>
<td>5.6.4.2 Circles in Robots</td>
<td>76</td>
</tr>
<tr>
<td>5.6.5 Rotating Station</td>
<td>78</td>
</tr>
<tr>
<td>5.6.6 Robot on a Rail</td>
<td>80</td>
</tr>
<tr>
<td>5.6.6.1 Robot on a Rail Setup - Static Rail Option</td>
<td>81</td>
</tr>
<tr>
<td>5.6.6.2 Robot on a Rail Setup - Dynamic Rail Option</td>
<td>81</td>
</tr>
<tr>
<td>5.6.6.3 XYZ or Relative Job Output Option</td>
<td>82</td>
</tr>
<tr>
<td>5.6.6.4 XYZ or Pulse Job Output Option</td>
<td>82</td>
</tr>
<tr>
<td>5.6.7 MotoCal Calibration Tasks For Offline Filter Creation</td>
<td>83</td>
</tr>
<tr>
<td>5.7 Troubleshooting G-Code to Robot Conversions</td>
<td>83</td>
</tr>
<tr>
<td>5.7.1 Finding G Code Errors</td>
<td>86</td>
</tr>
<tr>
<td>5.7.1.1 Determining G-Code Errors From Robot Job Errors</td>
<td>86</td>
</tr>
<tr>
<td>5.7.1.2 Dividing the G Code File</td>
<td>91</td>
</tr>
<tr>
<td>5.7.1.3 Quick Resolution of Robot Reach Issues</td>
<td>91</td>
</tr>
<tr>
<td>5.7.2 Log File</td>
<td>92</td>
</tr>
<tr>
<td>5.7.3 Job Conversion Reports</td>
<td>93</td>
</tr>
<tr>
<td>Chapter 6</td>
<td></td>
</tr>
<tr>
<td><strong>AutoCAD to Robot Job Converter</strong></td>
<td></td>
</tr>
<tr>
<td>6.1 Installing AutoCAD to Robot Job Converter</td>
<td>95</td>
</tr>
<tr>
<td>6.2 First Time Use</td>
<td>96</td>
</tr>
<tr>
<td>6.2.1 Sample Files</td>
<td>96</td>
</tr>
<tr>
<td>6.3 Understanding the AutoCAD to Robot Job Converter Work Area</td>
<td>97</td>
</tr>
<tr>
<td>6.3.1 Command Menus</td>
<td>97</td>
</tr>
<tr>
<td>6.3.2 Configuration Files Toolbar</td>
<td>98</td>
</tr>
<tr>
<td>6.3.2.1 XYZ or Relative Job Output Option</td>
<td>98</td>
</tr>
<tr>
<td>6.3.2.2 XYZ or Pulse Job Output Option</td>
<td>98</td>
</tr>
<tr>
<td>6.3.2.3 Template Files</td>
<td>99</td>
</tr>
<tr>
<td>6.3.2.4 Cell Configuration File</td>
<td>100</td>
</tr>
<tr>
<td>6.3.2.5 Job Configuration File</td>
<td>101</td>
</tr>
<tr>
<td>6.3.2.6 AutoCAD Configuration File</td>
<td>102</td>
</tr>
<tr>
<td>6.3.2.7 Clear Configuration Files</td>
<td>104</td>
</tr>
<tr>
<td>6.3.2.8 Reload Configuration Files</td>
<td>104</td>
</tr>
<tr>
<td>6.3.3 Converter Setup Toolbar</td>
<td>104</td>
</tr>
<tr>
<td>6.3.3.1 Select and Convert Points File</td>
<td>104</td>
</tr>
<tr>
<td>6.3.3.2 Play Job</td>
<td>104</td>
</tr>
</tbody>
</table>
Appendix A

Basic Introduction to G-Code

Appendix B

MotoCal Calibration Tasks For Offline Filter Creation

B.0.1 G-Code Editor Part Programming Tasks
B.0.2 Robot, Tool and Work piece Calibration
B.0.3 MotoSim EGEG Simulation Tasks - Part 2
B.0.4 Filter Jobs
Chapter 1
Introduction

The MotoSim Points Importer EG software suite is part of the MotoSoft™ family of software solutions. MotoSim Points Importer EG provides an easy way to create robot JBI job files using MotoSimEG models, XML schema files, and position data. Input position data can vary in format from standard text files, AutoCAD (optional) dwg files, standard G-Code files (optional) and Motoman standard XML job files. Additional modules (G-Code Converter and AutoCAD Converter) can be added to the suite to convert AutoCAD paths and G-code NC files to valid robot jobs.

The main MotoSim Points Importer EG application converts pre-defined XML files into robot JBI files. The XML files can be created using an XML editor or a CAD scripting language such as AutoCad® 2005 or Unigraphics®. The XML file must comply to the schema defined in Appendix C of this manual. MotoSim Points Importer EG supports robot motion commands and many mainstream INFORM III commands through the XML instructions. MotoSim Points Importer EG also provides the user a graphical user interface (GUI) for batch processing of files along with a .NET programming interface. Since MotoSim Points Importer EG is a client program of MotoSim EG, the user can view and play back newly created JBIs in a virtual environment before deploying the job to the real workcell.

The Points to JBI application is a new standard addition to the MotoSim Points Importer EG suite. It can be used to create JBI job files directly from standard ASCII files with comma or tab delimited point or coordinate information. This provides more direct control over the conversion process, making it easier to convert raw point files into the XML files required for the basic MotoSim Points Importer EG application.

Note: All MotoSim Points Importer EG converter applications require MotoSimEG to be installed on the same PC. Together, the converters and MotoSimEG, provide a comprehensive offline programming solution.
1.1 Conversion Process Overview

All position files go through a number of conversions. While these conversions happen transparently and automatically it is important to understand the intermediate file formats to help isolate problems when there are errors.

While the initial conversions are unique to each converter application, they do become uniform after the second conversion, with the output of a set of XML Job Files. The XML Job files:

a. Comply with the MotoSim Points Importer EG schema (see Appendix C)

b. Can be used (converted into a robot job) using the main MotoSim Points Importer EG application

Each converter application uses the main MotoSim Points Importer EG application to perform the third conversion. During this conversion, each XML Job file is converted into a MotoSimEG INF file. The fourth conversion is performed by MotoSimEG. During this conversion, the *.INF file is converted by MotoSimEG into a robot job or *.jbi file.
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 manual contains the following chapters:

**CHAPTER 1 - INTRODUCTION**
This chapter provides general information about the MotoSim Points Importer EG and its components, a list of reference documents, and customer service information.

**CHAPTER 2 - SAFETY**
This chapter provides information regarding the safe use and operation of the MotoSim Points Importer EG software.

**CHAPTER 3 - MOTOSIM POINTS IMPORTER EG**
This chapter provides information regarding the main MotoSim EG Points Importer application including descriptions of command menus, process buttons, and other key features of the user interface. Basic operations including; opening a cell, creating a .jbi robot job file, etc. and programming specifications for automation using Microsoft Visual Studio .Net. are also explained.

**CHAPTER 4 - POINTS TO ROBOT JOB CONVERTER**
This chapter provides detailed procedures for using the Points to Robot Job Converter application. The Points to Robot Job Converter provides greater control over the XML configuration files and conversion process than the standard MotoSim EG Points Importer program.

**CHAPTER 5 - G-CODE TO ROBOT JOB CONVERTER**
This chapter provides detailed procedures for using the G-Code to Robot Job Converter application including; installation, operation, and quick start guide.

**CHAPTER 6 - AUTOCAD TO ROBOT JOB CONVERTER**
This chapter provides detailed procedures for using the AutoCAD to Robot Job Converter application including; installation, operation, and quick start guide.

**APPENDIX A - BASIC INTRODUCTION TO G-CODE**
This chapter provides detailed procedures for using the AutoCAD to Robot Job Converter application including; installation, operation, and quick start guide.

**APPENDIX B - MOTOCAL CALIBRATION TASKS FOR OFFLINE FILTER CREATION**
This chapter provides detailed procedures for using the AutoCAD to Robot Job Converter application including; installation, operation, and quick start guide.
### 1.3 Features

Table 1 MotoSim Points Importer EG Features and Limitations

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Supported</th>
<th>Example</th>
<th>Miscellaneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point Data</td>
<td>Static Part with 2,3,4,5,6 axes</td>
<td>Yes</td>
<td>X, Y, Z, Rx, Rx, and R</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>Part of Rotary Table</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Robot</td>
<td>Coordinated Motion with Robots</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Coordinated Motion with Servo</td>
<td>No</td>
<td>SynchL, SynchC, SynchJ</td>
<td>1 Degree of Freedom -Rotational Axis</td>
</tr>
<tr>
<td></td>
<td>Positioners</td>
<td></td>
<td></td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>Coordinated Motion with Linear</td>
<td>Yes</td>
<td>Robot on a Rail</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tracks</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Template Files</td>
<td></td>
<td>No</td>
<td></td>
<td>Template files are used to provide</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>additional functionality. Generally they</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>are used to start processes like Arc On</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>or Arc Off, but can be used for any</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>process.</td>
</tr>
<tr>
<td>Inform</td>
<td>Joint Jobs</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Relative Jobs</td>
<td>Yes</td>
<td></td>
<td>Bug in MotoSim 2.5 and below requires 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>identical user frames</td>
</tr>
<tr>
<td></td>
<td>Joint Motion</td>
<td>Yes</td>
<td>MOVJ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Circular Motion</td>
<td>Yes</td>
<td>MOVC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Straight Line Motion</td>
<td>Yes</td>
<td>MOVL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Synchronous Motion</td>
<td>No</td>
<td>SynchL, SynchC, SynchJ</td>
<td></td>
</tr>
</tbody>
</table>
Table 2  G-Code Converter Features and Limitations

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Supported</th>
<th>Example</th>
<th>Miscellaneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNC</td>
<td>Static Part with CNC 2,3,4,5 Axes</td>
<td>Yes</td>
<td>X,Y,Z,A,B,C Supported</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Distance from Tool Base to rotational Axes with non-zero length</td>
<td>No</td>
<td>Robot and CNC compute tool lengths differently. CNC must set the Distance from Rotational Pivot point to tool base (or tool faceplate to rotational pivot point) to 0.</td>
<td>Note: This may be a limitation if customer does not use CNC source code but just the G-Code file.</td>
</tr>
<tr>
<td></td>
<td>Part of Rotary Table</td>
<td>Yes</td>
<td>X,Y,Z,A,B,C with A,B or C providing the Rotating Station Position</td>
<td></td>
</tr>
<tr>
<td>G-Code</td>
<td>Circle with I,J,K vectors</td>
<td>Yes</td>
<td>I,K vectors in G-Code define Circular Motion Center Point.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 Point 360 Circles</td>
<td>Yes</td>
<td>I,K circle vectors can be either relative or absolute coordinates.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I,J,K vectors as relative or Absolute Coordinates</td>
<td>Yes</td>
<td>I,K circle vectors can be either relative or absolute coordinates.</td>
<td></td>
</tr>
<tr>
<td>Robot</td>
<td>Coordinated Motion with Robots</td>
<td>No</td>
<td>1 degree of freedom - Rotational Axis Supported.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Coordinated Motion with Servo Positioners</td>
<td>Limited</td>
<td>SynchL, SynchC, SynchJ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Coordinated Motion with Linear Tracks</td>
<td>Yes</td>
<td>Robot on a Rail Supported</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Template Files</td>
<td>No</td>
<td>Template files are used to provide additional functionality. Generally they are used to start processes like Arc On or Arc Off, but can be used for any process.</td>
<td></td>
</tr>
</tbody>
</table>
### Table 2 G-Code Converter Features and Limitations

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Supported</th>
<th>Example</th>
<th>Miscellaneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inform</td>
<td>Joint Jobs</td>
<td>Yes</td>
<td></td>
<td>Bug in MotoSim 2.5 and below requires 2 identical user frames</td>
</tr>
<tr>
<td></td>
<td>Relative Jobs</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Joint Motion</td>
<td>Yes</td>
<td>MOVJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Circular Motion</td>
<td>Yes</td>
<td>MOVC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Straight Line Motion</td>
<td>Yes</td>
<td>MOVL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Synchronous Motion</td>
<td>Limited</td>
<td>SynchL, SynchC, SynchJ</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 3 AutoCAD Converter Features and Limitations

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Supported</th>
<th>Example</th>
<th>Miscellaneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoCAD</td>
<td>AutoCAD Full 2005, 2006, or 2008 versions</td>
<td>Yes</td>
<td></td>
<td>Note: One of these versions of AutoCAD must be on the same PC as AutoCAD Converter</td>
</tr>
<tr>
<td></td>
<td>Vector color defines Motion type</td>
<td>Yes</td>
<td></td>
<td>User can assign any AutoCAD color to any motion type (i.e. Red Vector = Linear Motion, White Vector = Joint Motion).</td>
</tr>
<tr>
<td></td>
<td>Template Files</td>
<td>Yes</td>
<td></td>
<td>Template files are used to provide additional functionality. Generally they are used to start processes like Arc On or Arc Off, but can be used for any process.</td>
</tr>
<tr>
<td>Robot</td>
<td>Coordinated Motion with Robots</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Coordinated Motion with Servo Positioners</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Coordinated Motion with Linear Tracks</td>
<td>No</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1.3.1 **Supported File Formats**

The type of converter to use is determined by the type of position data that is available to the user. The following chart shows the basic types of input data that can be converted into robot jobs.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Supported</th>
<th>Example</th>
<th>Miscellaneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inform</td>
<td>Joint Jobs</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Relative Jobs</td>
<td>Yes</td>
<td></td>
<td>Bug in MotoSim 2.5 and below require 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>identical user frames</td>
</tr>
<tr>
<td></td>
<td>Joint Motion</td>
<td>Yes</td>
<td>MOVJ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Circular Motion</td>
<td>Yes</td>
<td>MOVC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Straight Line Motion</td>
<td>Yes</td>
<td>MOVL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Synchronous Motion</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1.4 System Requirements

MotoSim Points Importer EG is a MotoSim EG client application and will run well within MotoSim EG’s minimum system requirements. MotoSim Points Importer EG requires an additional 25-30 megabytes of hard disk space.

- Operating System - Microsoft Windows XP
- MotoSim EG Viewer 6.4.2 or greater
- .NET Framework (2.0 preferred)
1.5 Installing MotoSim Points Importer EG

All MotoSoft software is provided on a single CD-ROM with a browser/installer utility (Motoman CD-ROM Browser, P/N 141720-1). Please refer to the CD Browser for detailed installation instructions.

MotoSim Points Importer EG is installed in c:\Program Files\Motoman\MotoSim Points Importer EG\.

To install MotoSim Points Importer EG, proceed as follows:

1. Insert CD Browser into your CD-ROM drive.
2. The setup program starts automatically.

Note: Setup starts automatically when the CD is inserted into the CD-ROM drive unless autoexecute has been disabled on your computer. If the setup program does not start, select demo32.exe from the CD-ROM Open Properties dialogue. Demoshield Player appears and prompts you for a .dbd file. Double-click *.dbd to begin the install process.

3. The introduction screen appears. Click Next to continue.

4. Click the Simulation Software button. The Motoman Software License Agreement page appears.
5. Click the Accept button to agree to the terms outlined in the scrolling window. The software selection screen appears.

6. Click the MotoSim Points Importer EG button to begin the installation process. The MotoSim Points Importer EG install wizard begins.

7. Follow the MotoSim Points Importer EG install wizard instructions as it guides you through the installation process.

1.5.1 Installing the Hardware Key

The USB hardware key supplied with MotoSim Points Importer EG must be installed on your computer or MotoSim Points Importer EG will not function properly. The hardware key attaches to the computer’s USB port. The USB port is commonly used to run peripheral devices on your computer. To attach the hardware key to the USB port, proceed as follows:

1. Locate an open USB port on your computer.
2. Carefully insert the hardware key into the USB port. If the key does not fit, do not force it. The key should fit snugly but does not require significant force to insert.

1.6 First Time Use

Before you begin using MotoSim EG Points Importer, you must first have a valid MotoSim EG .cel robot cell file and XML file using the correct schema. Several sample files are included with MotoSim Points Importer EG and can be located in c:\Program Files\Motoman\MotoSim Points Importer EG\Cells.

Note: If your XML positions are defined in terms of a User Frame or a MotoSim EG Model, you must have the work frame defined in the MotoSim EG cell file before using it with MotoSim EG Points Importer.

The first time you run MotoSim EG Points Importer you will need to open a valid MotoSim EG .mcl cel file. After first time use, MotoSim EG Points Importer will automatically start MotoSim EG and open the last cell modified.
1.7 Learning MotoSim Points Importer EG

Motoman provides a variety of options to help you to learn MotoSim Points Importer EG, including online Help, training and technical support. Access to a host of continually updated Web resources for learning MotoSim Points Importer EG, including technical support information, can be found at the Motoman web site (www.motoman.com).

1.7.1 Using Help

The MotoSim Points Importer EG application includes complete documentation in an HTML-based help system. It contains essential information on using all of the MotoSim Points Importer EG commands, features and tools. The HTML format provides easy navigation. Individual topics can also be printed out to provide a handy desktop reference.

To properly view Help topics, you need MicroSoft® Internet Explorer 6.0 (or later).

The Help assumes you have a working knowledge of your computer and its operating conventions, including how to use a mouse and standard menus and commands. It also assumes you know how to open, save, and close files. For help with any of these techniques, please see your MicroSoft Windows documentation.

1.7.2 Motoman Technical Education Center (MTEC)

The Motoman Technical Education Center offers over 56 different courses including Basic Programming, Advanced Programming, Maintenance, Concurrent I/O, Purchasable Options, and Customized Training. Web-based or Computer-based training is also available for selected topics.

Motoman training courses provide classroom instruction combined with hands-on training (normally 2-student-per-robot ratio). Class size is limited to provide a more effective and enhanced learning environment.

With three U.S. facilities and more than 60 training robots, Motoman offers the most complete training package in the industry. Since MTEC is an authorized provider for the International Association for Continuing Education and Training (IACET), each student who receives at least 70% on the final exam will receive Continuing Education Units (CEUs). These CEUs are transferable college credits, which are awarded based on contact hours of the course. The Motoman Technical Education Center is the first robotic training facility with IACET accreditation and is also approved for training of veterans.

Motoman offers training at our headquarters in West Carrollton, Ohio and at our remote facilities in Irvine, California; Wixom, Michigan; Mississauga, Canada; and Aguascalientes, Mexico. Motoman also performs On-Site training at customer sites if required but recommends training at Motoman since the environment is more conducive to effective learning.

For more information, visit our website (http://www.motoman.com/support/training/training.htm).
1.7.3 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/xml)
XML.org - (http://www.xml.org)

1.8 Reference to Other Documentation

For additional information, refer to the following:

• MotoSim EG Instruction Manual (P/N 152002-1)
• Motoman Manipulator Manual for your robot model
• Motoman Operator’s Manual for your application

1.9 Customer Service Information

If you are in need technical assistance, contact the Motoman service staff at (937) 847-3200. Please have the following information ready before you call:

• MotoSim EG Points Importer version
• MotoSim EG version
• Operating system (Windows 2000, Windows XP)
• System configuration (hard disk capacity, memory, software, etc.)
• Description of difficulty (note any error messages)
Chapter 2
Safety

2.1 Introduction

It is the purchaser’s responsibility to ensure that all local, county, state, and national codes, regulations, rules, or laws relating to safety and safe operating conditions for each installation are met and followed.

We suggest that you obtain and review a copy of the ANSI/RIA National Safety Standard for Industrial Robots and Robot Systems. This information can be obtained from the Robotic Industries Association by requesting ANSI/RIA R15.06. The address is as follows:

Robotic Industries Association
900 Victors Way
P.O. Box 3724
Ann Arbor, Michigan 48106
TEL: (734) 994-6088
FAX: (734) 994-3338

Ultimately, the best safeguard is trained personnel. The user is responsible for providing personnel who are adequately trained to operate, program, and maintain the robot cell. The robot must not be operated by personnel who have not been trained!

We recommend that all personnel who intend to operate, program, repair, or use the robot system be trained in an approved Motoman training course and become familiar with the proper operation of the system.
This safety section addresses the following:

- Standard Conventions (Section 2.2)
- General Safeguarding Tips (Section 2.3)
- Mechanical Safety Devices (Section 2.4)
- Installation Safety (Section 2.5)
- Programming Safety (Section 2.6)
- Operation Safety (Section 2.7)
- Maintenance Safety (Section 2.8)

2.2 Standard Conventions

This manual includes information essential to the safety of personnel and equipment. As you read through this manual, be alert to the four signal words:

DANGER!

WARNING!

CAUTION!

NOTE:

Pay particular attention to the information provided under these headings which are defined below (in descending order of severity).

⚠️ DANGER!
Information appearing under the DANGER caption concerns the protection of personnel from the immediate and imminent hazards that, if not avoided, will result in immediate, serious personal injury or loss of life in addition to equipment damage.

⚠️ WARNING!
Information appearing under the WARNING caption concerns the protection of personnel and equipment from potential hazards that can result in personal injury or loss of life in addition to equipment damage.

⚠️ CAUTION!
Information appearing under the CAUTION caption concerns the protection of personnel and equipment, software, and data from hazards that can result in minor personal injury or equipment damage.

-note:
Note: Information appearing in a Note caption provides additional information which is helpful in understanding the item being explained.
2.3 **General Safeguarding Tips**

All operators, programmers, plant and tooling engineers, maintenance personnel, supervisors, and anyone working near the robot must become familiar with the operation of this equipment. All personnel involved with the operation of the equipment must understand potential dangers of operation. General safeguarding tips are as follows:

- Improper operation can result in personal injury and/or damage to the equipment. Only trained personnel familiar with the operation of this robot, the operator's manuals, the system equipment, and options and accessories should be permitted to operate this robot system.
- Do not enter the robot cell while it is in automatic operation. Programmers must have the teach pendant when they enter the robot cell.
- Improper connections can damage the robot. All connections must be made within the standard voltage and current ratings of the robot I/O (Inputs and Outputs).
- The robot must be placed in Emergency Stop (E-STOP) mode whenever it is not in use.
- In accordance with ANSI/RIA R15.06, section 6.13.4 and 6.13.5, use lockout/tagout procedures during equipment maintenance. Refer also to Section 1910.147 (29CFR, Part 1910), Occupational Safety and Health Standards for General Industry (OSHA).

2.4 **Mechanical Safety Devices**

The safe operation of the robot, positioner, auxiliary equipment, and system is ultimately the user's responsibility. The conditions under which the equipment will be operated safely should be reviewed by the user. The user must be aware of the various national codes, ANSI/RIA R15.06 safety standards, and other local codes that may pertain to the installation and use of industrial equipment. Additional safety measures for personnel and equipment may be required depending on system installation, operation, and/or location. The following safety measures are available:

- Safety fences and barriers
- Light curtains
- Door interlocks
- Safety mats
- Floor markings
- Warning lights

Check all safety equipment frequently for proper operation. Repair or replace any non-functioning safety equipment immediately.

2.5 **Installation Safety**

Safe installation is essential for protection of people and equipment. The following suggestions are intended to supplement, but not replace, existing federal, local, and state laws and regulations.
Additional safety measures for personnel and equipment may be required depending on system installation, operation, and/or location. Installation tips are as follows:

- Be sure that only qualified personnel familiar with national codes, local codes, and ANSI/RIA R15.06 safety standards are permitted to install the equipment.
- Identify the work envelope of each robot with floor markings, signs, and barriers.
- Position all controllers outside the robot work envelope.
- Whenever possible, install safety fences to protect against unauthorized entry into the work envelope.
- Eliminate areas where personnel might get trapped between a moving robot and other equipment (pinch points).
- Provide sufficient room inside the workcell to permit safe teaching and maintenance procedures.

2.6 Programming Safety

All operators, programmers, plant and tooling engineers, maintenance personnel, supervisors, and anyone working near the robot must become familiar with the operation of this equipment. All personnel involved with the operation of the equipment must understand potential dangers of operation. Programming tips are as follows:

Any modifications to PART 1 of the XRC controller PLC can cause severe personal injury or death, as well as damage to the robot! Do not make any modifications to PART 1. Making any changes without the written permission of Motoman will VOID YOUR WARRANTY!

Some operations require standard passwords and some require special passwords. Special passwords are for Motoman use only. YOUR WARRANTY WILL BE VOID if you use these special passwords.

Back up all programs and jobs onto a floppy disk whenever program changes are made. To avoid loss of information, programs, or jobs, a backup must always be made before any service procedures are done and before any changes are made to options, accessories, or equipment.

The concurrent I/O (Input and Output) function allows the customer to modify the internal ladder inputs and outputs for maximum robot performance. Great care must be taken when making these modifications. Double-check all modifications under every mode of robot operation to ensure that you have not created hazards or dangerous situations that may damage the robot or other parts of the system.

- Improper operation can result in personal injury and/or damage to the equipment. Only trained personnel familiar with the operation, manuals, electrical design, and equipment interconnections of this robot should be permitted to operate the system.
- Inspect the robot and work envelope to be sure no potentially hazardous conditions exist. Be sure the area is clean and free of water, oil, debris, etc.
- Be sure that all safeguards are in place.
- Check the E-STOP button on the teach pendant for proper operation before programming.
- Carry the teach pendant with you when you enter the workcell.
• Be sure that only the person holding the teach pendant enters the workcell.
• Test any new or modified program at low speed for at least one full cycle.

2.7 Operation Safety

All operators, programmers, plant and tooling engineers, maintenance personnel, supervisors, and anyone working near the robot must become familiar with the operation of this equipment. All personnel involved with the operation of the equipment must understand potential dangers of operation. Operation tips are as follows:

• Be sure that only trained personnel familiar with the operation of this robot, the operator's manuals, the system equipment, and options and accessories are permitted to operate this robot system.
• Check all safety equipment for proper operation. Repair or replace any non-functioning safety equipment immediately.
• Inspect the robot and work envelope to ensure no potentially hazardous conditions exist. Be sure the area is clean and free of water, oil, debris, etc.
• Ensure that all safeguards are in place.
• Improper operation can result in personal injury and/or damage to the equipment. Only trained personnel familiar with the operation, manuals, electrical design, and equipment interconnections of this robot should be permitted to operate the system.
• Do not enter the robot cell while it is in automatic operation. Programmers must have the teach pendant when they enter the cell.
• The robot must be placed in Emergency Stop (E-STOP) mode whenever it is not in use.
• This equipment has multiple sources of electrical supply. Electrical interconnections are made between the controller, external servo box, and other equipment. Disconnect and lockout/tagout all electrical circuits before making any modifications or connections.
• All modifications made to the controller will change the way the robot operates and can cause severe personal injury or death, as well as damage the robot. This includes controller parameters, ladder parts 1 and 2, and I/O (Input and Output) modifications. Check and test all changes at slow speed.

2.8 Maintenance Safety

All operators, programmers, plant and tooling engineers, maintenance personnel, supervisors, and anyone working near the robot must become familiar with the operation of this equipment. All personnel involved with the operation of the equipment must understand potential dangers of operation. Maintenance tips are as follows:

• Do not perform any maintenance procedures before reading and understanding the proper procedures in the appropriate manual.
• Check all safety equipment for proper operation. Repair or replace any non-functioning safety equipment immediately.

• Improper operation can result in personal injury and/or damage to the equipment. Only trained personnel familiar with the operation, manuals, electrical design, and equipment interconnections of this robot should be permitted to operate the system.

• Back up all your programs and jobs onto a floppy disk whenever program changes are made. A backup must always be made before any servicing or changes are made to options, accessories, or equipment to avoid loss of information, programs, or jobs.

• Do not enter the robot cell while it is in automatic operation. Programmers must have the teach pendant when they enter the cell.

• The robot must be placed in Emergency Stop (E-STOP) mode whenever it is not in use.

• Be sure all safeguards are in place.

• Use proper replacement parts.

• This equipment has multiple sources of electrical supply. Electrical interconnections are made between the controller, external servo box, and other equipment. Disconnect and lockout/tagout all electrical circuits before making any modifications or connections.

• All modifications made to the controller will change the way the robot operates and can cause severe personal injury or death, as well as damage the robot. This includes controller parameters, ladder parts 1 and 2, and I/O (Input and Output) modifications. Check and test all changes at slow speed.

• Improper connections can damage the robot. All connections must be made within the standard voltage and current ratings of the robot I/O (Inputs and Outputs).
Chapter 3
MotoSim Points Importer EG

This chapter provides basic operating instructions for the main MotoSim Points Importer EG application.

3.1 Understanding the MotoSim Points Importer EG Work Area

MotoSimEG Points Importer features an easy to use graphical interface. All button commands can also be found under the File dropdown window. In addition, all MotoSimEG Points Importer commands can be incorporated into your own automation software using Microsoft® Visual Studio.NET®.
3.2 Command Menus

File
Contains all MotoSimEG Points Importer process commands as listed below. Allows the user to open and close MotoSimEG Cell files and process or convert XML Job Files into a robot job.

Utility
Contains the Batch Process command for processing multiple XML job files at one time (see Section 3.6.2, "Batch Job Processing").

Points to JBI
The Points to Robot Job Converter application converts position data into robot JBI files using user defined configuration files. This is a quick method of converting raw position data into a robot Job.

G-Code to JBI (Optional)
The G-Code to JBI application is an optional productivity add-on to MotoSimEG Points Importer. This application converts CNC G Code programs into robot jobs. User defined XML configuration files are used to control the conversion process. Refer to Section 5 for detailed instructions.

AutoCAD to JBI (Optional)
Opens the AutoCAD to Robot Job Converter application. The AutoCAD to Robot Job Converter application is an optional productivity add-on that enables you to import MotoSimEG process paths from AutoCAD. This allows you to create robot jobs directly from AutoCAD path models.

Note: AutoCAD 2005, 2006 or 2008 Full Version must be installed on the same PC with the Point Importer EG Software Suite.
Refer to Section 6 for detailed instructions.

Help
• About - Contains version information.
• Contents - Opens the interactive Help file.

3.3 Process Buttons

Open Cell
Starts the MotoSimEG simulation software and opens the selected MotoSim EG cell file. The MotoSimEG cell name is displayed in the MotoSim EG File Status window along with its current status.

Close Cell
Closes the active MotoSimEG cell file.

Open XML
Opens the selected XML positions file. The XML file name is displayed in the XML File Status window along with its current status.

Create Model
Creates a MotoSimEG Axis6 model file based on the information provided in the XML positions file. The MotoSim EG model file name is displayed in the Model Status window along with its current status. The new model file is stored in the c:\Program Files\Motoman\MotoSimEG Points Importer\Cells folder.

Note: The default MotoSim EG Model file name is based on the XML file used. This name can be changed by simply clicking in the Model Name window and typing the desired name.
Create JBI
Creates a valid Motoman .jbi robot job file based on the MotoSimEG model file created. The Motoman JBI file name is displayed in the JBI Status window along with its current status. The new job file is stored in the c:\Program Files\Motoman\MotoSimEG Points Importer\Cells directory in the appropriate robot file.

Note: The default JBI file name is based on the XML file used. This name can be changed by simply clicking in the Job Name window and typing the desired name.

Play JBI
Test runs the active job file in MotoSimEG and displays the results in the Play Status window.

Edit XML
Opens the XML positions file in the default XML editor.

Edit INF
Opens the INF file in the default text editor.

Edit JBI
Opens the robot JBI file in the default text editor.

Job Dir.
Opens the MotoSimEG directory where the robot jobs are created.

3.4 Status Windows

MotoSim EG File Status
Displays the currently active MotoSimEG cell file and its status.

XML File Status
Displays the currently active XML positions file and its status.

Model Status
Displays the name of the created MotoSimEG .mdl model file and its status.

JBI Status
Displays the name of the created Motoman .jbi robot job file and its status.

Play Status
Displays the playback status of the current robot job file.

3.5 Transparency

The transparency slider allows you to make the MotoSimEG Points Importer work area appear transparent. When this is done, other application windows can be seen through the work area. This can allow you to view a robot job running in the MotoSimEG application window while continuing work in the MotoSimEG Points Importer work area.
### 3.6 Working with MotoSim Points Importer EG

#### 3.6.1 Creating a Job

1. Click the [Open Cell] button. The Open MotoSimEG Cell Files window appears.
2. Select the desired .cel file and click Open. The MotoSimEG application starts and opens the selected file.
3. From the MotoSimEG Points Importer application, click the [Open XML] button. The Open XML Position Files window appears.
4. Select the desired XML position file and click Open. The file name appears in both the Model Name and Job Name windows.

   *Note: To change Job name, simply click in the Job Name window and type the desired name.*

5. Click the [Create Model] button.
6. Click the [Create JBI] button. Points Importer creates a valid .jbi robot job file and stores it in the c:\Program Files\Motoman\MotoSimEG Points Importer\Cells directory in the appropriate robot file.

#### 3.6.2 Batch Job Processing

1. Click the [Open Cell] button to open a valid MotoSimEG .cel file. The Open MotoSimEG Cell Files window appears.
2. Select the desired .mcl file and click [Open]. MotoSimEG starts and opens the selected file.

4. Click the [Add] button to select input XML files.
5. Click the [Clear] button to remove files from list.
6. Click the [Process] button. Points Importer creates the valid .jbi robot job files and stores them in the c:\Program Files\Motoman\MotoSimEG Points Importer\Cells directory in the appropriate robot folder.
Chapter 4
Points to Robot Job Converter

The Points to Robot Job Converter application is intended as a quick and easy method of converting standard ASCII files with comma or tab delimited point or coordinate information into robot JBI job files. Point data can originate from many sources such as CAD or CAD/CAM applications, NotePad, or Excel and Access database exports.

The figure below shows an overview of the Points to Robot Job process including intermediate files that are produced.

The points file is first converted into a master XML file. This file is then broken into master and callable subroutines XML job files. Each of these files has both the XML header section as well as the point location data. These XML files are now compatible with the main MotoSim EG Points Importer EG application.

The third and fourth conversions are triggered by the Points to Robot Job Converter application but are actually performed by the Points Importer EG and the MotoSim EGE applications. These conversions produce an intermediate set of INF files that are used by MotoSim EGE to produce the final set of robot JBI job files.
4.1 Understanding the Points to Robot Job Converter Work Area

The Points to Robot Job Converter application features an easy to use graphical interface. All toolbar button commands can also be found under the File dropdown window. In addition, you can toggle toolbars on and off using the View menu commands.

4.1.1 Command Menus

File
Contains all Points to Robot Job Converter process commands as listed below.

View
Allows you to toggle the view for each toolbar.

Help
- About - Contains version information.
- Contents - Opens the interactive Help file.
4.1.2 Configuration Files Toolbar

The Configuration Files toolbar enables you to determine what Cell, Job, and Points configuration files will be used to create the robot job file. Each window is color coded to show the status of each file. Red indicates that there is a problem with this file. Yellow indicates that the file has yet to be loaded, and Green means that all files are correct and the application is ready to convert your points file. You can also click on the wrench icon to edit any of the configuration files.

WARNING!
Changes to configuration files do not take effect until files are reloaded.

4.1.2.1 Cell Configuration File

The xml cell configuration file contains data related to the robot and the MotoSim EG cel file being used for the conversion. Allowable fields and values are described in the CelConfigurationSchema(x).xsd (where (x) is an integer that is incremented each time the schema file is updated). Refer to Appendix D for detailed XML schema.

**Sample Cell Configuration File:**

```
- <Root>
  - <Settings>
    - <ArmConfig>
      <S>24728</S>
      <L>24728</L>
      <U>24728</U>
      <R>24728</R>
      <B>24728</B>
      <T>24728</T>
    - </ArmConfig>
  - </Settings>
- </Root>
```

The Cell Configuration File window allows you to upload and edit the XML cell configuration file you want to use to create your robot job.

**To load a cell configuration file:**

1. Click on the folder icon.
2. Locate the file you want to use and click Open. If Points to Robot Job Converter displays the cell file in Red, there may be a problem with the selected file. Make certain you have selected the correct XML cell configuration file or edit the file to correct the problem.

**To edit a cell configuration file:**

1. Click on the wrench icon.
2. The Points to Robot Job Converter application opens the XML file in the default XML editor. Remember: Changes to configuration files do not take effect until all file are reloaded.
3. Save your changes to the cell configuration file.
4. Click on the All folder to reload the configuration files and allow changes to take effect.
4.1.2.2 Job Configuration File

The xml Job Configuration file contains information that describes how the point job will be converted into a robot job. This file defines a number of point related data such as:

- Dimensional units of the data in the point file
- Angle System which defines the robot Orientation Coordinate System
- Default Motion types and speeds

This file also contains Cell Data that pertains to the how the Points are converted. Example data includes:

- User Frame to attach the data to
- Maximum Number of Lines in each Subroutine
- Job Output Options (Relative or Joint)
- Tool Number

Allowable fields and values are described in the JobConfigSchemaV(x).xsd (where (x) is an integer that is incremented each time the schema file is updated). Refer to Appendix D for detailed XML schema).

Sample Job Configuration File:

```
<Root>
  <Settings>
    <InformFileFormats MasterandSubs="True" MasterName="gmaster" MaxLines="900"/>
    <AngleSystem>YPR</AngleSystem>
    <Units>
      <LengthUnits>Inches</LengthUnits>
      <AngleUnits>Degrees</AngleUnits>
      <TimeUnits>Minutes</TimeUnits>
    </Units>
    <DefaultVelocity>
      <Linear>50.0</Linear>
      <Joint>5</Joint>
    </DefaultVelocity>
    <DefaultMotionType>Linear</DefaultMotionType>
    <ReferenceModel>UF1</ReferenceModel>
    <Tool>0</Tool>
  </Settings>
</Root>
```

The Job Configuration File window allows you to upload and edit the XML job configuration file you want to use to create your robot job.

To load a job configuration file:
1. Click on the folder icon.
2. Locate the file you want to use and click Open. If Points to Robot Job Converter displays the cell file in Red, there may be a problem with the selected file. Make certain you have selected the correct XML job configuration file or edit the file to correct the problem.

To edit a job configuration file:
1. Click on the wrench icon.
2. The Points to Robot Job Converter application opens the XML file in the default XML editor. Remember: Changes to configuration files do not take effect until all file are reloaded.
3. Save your changes to the job configuration file.
4. Click on the All folder to reload the configuration files and allow changes to take effect.

### 4.1.2.3 Points Configuration File

The Points Configuration file contains regular expressions (RegEx) that tell the software how to parse or extract the position data from the ASCII file. This file is setup by Motoman and users are generally not required to modify it. Please contact Motoman if discover errors with a specific ASCII File format.

Allowable fields and values are described in the PointsConverterSchema\(V(x)\).xsd (where \(x\) is an integer that is incremented each time the schema file is updated). Refer to Appendix for details.

**Sample Job Configuration File:**

```xml
<Root>
  - <Root>
    - <Settings>
      - <RegExParseInt>(\[\W*[-+]*\d{1,3}\..\d{0,4}\s{0,5}]*</RegExParseInt>
      - <RegExParseInt>((\[+-\])*\d{0,6}\..*\d{0,6})*</RegExParseInt>
      - <AutoReadLastPointsFile>False</AutoReadLastPointsFile>
      - <AutoReadLastConfigFiles>True</AutoReadLastConfigFiles>
      - <InformFileFormats MaxLines="750" />
    </Settings>
</Root>
```

The Points Configuration File window allows you to upload and edit the XML points configuration file you want to use to create your robot job.

**To load a points configuration file:**

1. Click on the folder icon.
2. Locate the file you want to use and click Open. If Points to Robot Job Converter displays the cell file in Red, there may be a problem with the selected file. Make certain you have selected the correct XML points configuration file or edit the file to correct the problem.

**To edit a job configuration file:**

1. Click on the wrench icon.
2. The Points to Robot Job Converter application opens the XML file in the default XML editor. Remember: Changes to configuration files do not take effect until all file are reloaded.
3. Save your changes to the points configuration file.
4. Click on the All folder to reload the configuration files and allow changes to take effect.

### 4.1.2.4 Clear Configuration Files

The Clear Configuration Files button unloads all your cell configuration file selections from application memory.

### 4.1.2.5 Reload Configuration Files

The Reload Configuration Files button reloads all configuration files. It is very important to remember to reload configuration files after making any changes to the configurations files. This updates the configuration files allowing any changes to take effect in the next conversion cycle.
4.1.3 **Converter Setup Toolbar**

The Converter Setup toolbar enables you to select the points file you wish to convert to a robot job. Once the job is converted, you can also play the job in MotoSim EG.

4.1.3.1 **Select and Convert Points File**

The Points File button allows you to select the points file you want to convert to a robot job.

**To select and convert a points file:**

1. Click on the Points File button.
2. Locate the file you want to convert and click Open. Points to Robot Job Converter begins processing the files you have selected to create a robot JBI file.
3. Upon success, you will receive a message stating that the points file has been successfully converted into a robot job. Click OK.
4. You may now play the converted job in MotoSim EG.

4.1.4 **Edit Toolbar**

Converting a points file to a robot job creates several intermediate process files including XML and INF. The Edit toolbar allows you to open and view each of these intermediate files generated by Points to Robot Job Converter. This can be a valuable tool for troubleshooting your job conversion. You can also view any error files or open the job folder directly to view other files associated with your project.

4.1.4.1 **Points File**

**To view/edit the points file:**

1. Click on the Points File button.
2. The Points to Robot Job Converter application opens the Points file in the default text editor.
3. Save your changes to the points configuration file.
4.1.4.2 XML Files

The Points to Robot Job Converter application creates several XML files during the conversion process. Point data is converted into an intermediate XML format creating a Master file and several subroutine file that will ultimately be converted into the .jbi job format.

To view/edit the XML files:
1. Select the XML file from the dropdown menu.
2. Click on the XML Files button.
3. The Points to Robot Job Converter application opens the XML file in the default XML editor.
4. Save your changes to the XML file.

4.1.4.3 INF Files

The Points to Robot Job Converter application creates one INF files for each xml job file created. The intermediate XML files are converted into the INF format that can be converted by the MotoSim EG application into the .jbi job format.

To view/edit the INF files:
1. Select the INF file from the dropdown menu.
2. Click on the INF Files button.
3. The Points to Robot Job Converter application opens the INF file in the default text editor.
4. Save your changes to the INF file.

4.1.4.4 JBI Files

To view/edit the JBI files:
1. Select the JBI file from the dropdown menu.
2. Click on the JBI Files button.
3. The Points to Robot Job Converter application opens the JBI file in the default text editor.
4. Save your changes to the JBI file.

4.1.4.5 View Error File

The View Error File button opens any error file generated by the Points to Robot Job Converter application during the job conversion process.

>Note: This file is only created when an error occurs in the INF to JBI conversion. Most errors are typically related to robot reach.

4.1.4.6 Job Directory

The JOB DIR. button allows you to browse to and open the job folder directly to view any file associated with your project.
4.1.5 **Transparency**

The transparency slider allows you to make the MotoSim Points Importer EG work area appear transparent. When this is done, other application windows can be seen through the work area. This can allow you to view a robot job running in the MotoSim EG application window while continuing work in the MotoSim Points Importer EG work area.

4.1.5.1 **On Top Check Box**

The On Top check box allows you to set the Points to Robot Job Converter application to always appear on top of other tiled windows or applications.

⚠️ **CAUTION!**
This Feature may make it difficult to see certain error messages or forms. Use with caution.

4.1.6 **Fault Logging**

When Fault Logging is ON, detailed event messages are written to the Windows Event Log - MotomanPointsImporterWizardC - as shown below.

Note: Only events from the last conversion are listed. When a new conversion starts, old conversion events are erased.

![Event Viewer](image)

This information can be saved and emailed to Motoman support personal to help troubleshoot various position data to robot job conversion errors.

**To save the log file:**

1. Select Action > Save Log File As... from the dropdown menu.
2. Enter the name you want to save the file as and click OK.
4.2 Opening Points to Robot Job Converter

The Points to Robot Job Converter application is accessed directly from the MotoSim Points Importer EG command menu.

4.3 Getting Started (Quick Start)

This section provides a quick overview describing how to convert a new points file to a robot jbi job files using the Points to Robot Job Converter application. Before using the Points to Robot Job Converter application, you must first have a valid MotoSim EG .cel robot cell file and user-defined XML configuration files using the correct schema. Several sample files are included with MotoSim Points Importer EG and can be located in c:\Program Files\Motoman\MotoSim Points Importer EG\Cells\Points Demo.
4.3.1 MotoSim EG Cell Setup

Before you can convert any points files, you must have a MotoSim EG cell file created. This cell must include a robot, Tool Center Point (TCP) or TCP set, and a user frame (part) coordinate system setup.

To create a new MotoSim EG Cell:

1. From the MotoSim EG main window, select File > New Cel Project... The New Cel window appears.

2. Enter the desired name for the cell in the File Name window and click Open. The new cell is created in MotoSim EG.

To add a robot to the cell file:

1. With the new cell open in MotoSim EG, select Robot > Add Robot... The Open window appears.

2. Browse to the desired robot folder and select the All.prm file and click Open. The Install Robot window appears.

3. If this is the correct robot, click OK. The robot model appears in the cell.

To add TCP offsets:

1. With the cell open and robot model visible, zoom into the T-axis of the robot.

2. Select Robot > Data Setting > Tool Data. The TOOL Editor window appears.

3. Modify the X, Z, and Ry values to reflect your tool.
To add Tool Model (Optional):

An optional tool model can be added to the cell to act as a visual aid for the TCP location.

1. From the MotoSim EG tool bar, select the CAD Tree button. The CAD Tree window appears.
2. With the tree expanded, right click on the robot tcp and select New Model... from the dropdown window. The Add Model Dialog window appears.
3. Enter the name of the tool model and click OK.
4. Click OK again to confirm file creation. The tool model appears in the CAD tree.
5. Double click on the new tool model file in the CAD tree. The tool model edit window appears.

6. From the Add Parts dropdown window, select CONE and click Add. The CONE Edit window appears.
7. Modify the L. Diameter to reduce the diameter of the cone and click OK.
8. Close the Tool Model window.
9. From the CAD Tree, right click on the tool model and select Select Position... from the dropdown window. The Position window appears.

![Position Tool Model]

10. Change the Z(mm) value to -100 to place the tip of the cone at the tool frame origin and click OK.
11. Close the CAD Tree view.

**To create a User Frame Model (Optional):**

1. From the MotoSim EG tool bar, select Robot > Reach View > Reach View. The Reach Area window appears.

![Reach Area]

2. Select the Tool radio button from the Mode panel.
3. Click Execute.
4. Click OK.
5. With the CAD tree expanded, right click on world and select New Model... from the dropdown window. The Add Model Dialog window appears.
6. Enter the name of the user frame model and click OK.
7. Click OK again to confirm file creation. The user frame model appears in the CAD tree.
8. Double click on the new user frame model file in the CAD tree. The user frame model edit window appears.
9. From the Add Parts dropdown window, select BOX and click Add. The BOX Edit window appears.
10. Modify the width and height to reduce the size of the box and click OK.
11. Close the User Frame Model window.
12. From the CAD Tree, right click on the user frame model and select Select Position... from the dropdown window. The Position window appears.

13. Change the X (mm) and Z(mm) values to place the user frame model in front of the robot model and within the robot reach.
14. Click OK.
15. Close the CAD Tree view.
16. From the MotoSim EG tool bar, select Robot > Reach View > Delete Reach View.

To define the user frame:
1. From the MotoSim EG tool bar, select Robot > Data Setting > Set User Frame... The Set User Frame window appears.

2. Using the cursor, select the corner of the user frame model.
3. De-select the Active radio button and click Close.

To save the cell file:
1. From the MotoSim EG tool bar, select File > Save.
4.3.2 Cell Configuration Setup

To setup initial arm configuration in XML configuration file:

1. From the MotoSim EG main window, click on the Position and Job View button from the Main tool bar. The Position and Job windows appear.

2. From the Position window, click on the Frame button.

3. Select the Pulse radio button from the Display Reference Frame window and click OK.

4. Locate the ArmConfig element in the XML Configuration file.

   <ArmConfig>
   <S>-6733</S>
   <L>118307</L>
   <U>87216</U>
   <R>-20336</R>
   <B>13994</B>
   <T>10689</T>
   </ArmConfig>

5. Copy and Paste the pulse position data for each of the axes (SLURBT) into the S, L, U, R, B and T child elements.
4.3.3 Job Configuration Setup

To set units in Job configurations file:
The measurement units used in the XML Configuration file must match those in the points file.

1. Open the XML Configuration file in an XML editor or text editor such as Notepad.
2. Locate the Units element.
   ```xml
   <Units>
   <LengthUnits>Millimeters</LengthUnits>
   <AngleUnits>Degrees</AngleUnits>
   <TimeUnits>Minutes</TimeUnits>
   </Units>
   ```
3. Change the LengthUnits element to either Millimeters or Inches.
4. Change the AngleUnits element to either Degrees or Radians.
5. Change the TimeUnits element to either Minutes or Seconds.

To setup user frame in XML configuration file:
1. Locate the ReferenceModel element and record the User Frame (UF1-UF10).
   ```xml
   <ReferenceModel>UF1</ReferenceModel>
   ```

To make the output job a relative job (Cartesian coordinates), add the following commands to the configuration file:

All RobotPosture commands should be set to 0. MotoSimEG will automatically assign them the correct values.

```xml
<JobOutputOptions>
   <PositionConfiguration>
      <UserFrame>1</UserFrame>
   </PositionConfiguration>
   <RobotPosture>
      <FlipPosture>0</FlipPosture>
      <UpperPosture>0</UpperPosture>
      <FrontPosture>0</FrontPosture>
      <RPosture>0</RPosture>
      <TPosture>0</TPosture>
   </RobotPosture>
</JobOutputOptions>
```

Note: All RobotPosture commands should be set to 0. MotoSimEG will automatically assign them the correct values.
To make the output job a joint job:

1. Locate the JobOutputOptions command and add a PulseJobType Command with a True data field as shown below.

2. Or delete the JobOutputOptions command and sub commands for the configuration file.

```xml
<JobOutputOptions>
  <JobType><PulseJobType>True</PulseJobType></JobType>
</JobOutputOptions>
```

### 4.3.4 Points Configuration Setup

**To setup maximum number of points in a single subroutine:**

1. Locate the InformFileFormats element and the MaxLines attribute and a number between 100 and 1000

```xml
<InformFileFormats = MaxLines="900"/>
```

**To add a new the Regular Expression:**

1. Locate the exiting RegExParseInt elements

2. Create a new RegExParseInt element and insert after the last RegExParseInt element

```xml
<RegExParseInt>((\d{1,3}\.\d{0,4})\s{0,5})*</RegExParseInt>
<RegExParseInt>(([+-])\d{0,6}\.*\d{0,6})*</RegExParseInt>
```

**To control automatic loading of configuration files:**

1. Locate the exiting AutoReadLastConfigFiles and the element

2. Set it to True to automatically read all three configuration files the next time the Point to JOB Form is loaded.

```xml
<AutoReadLastConfigFiles>True</AutoReadLastConfigFiles>
```

**To modify an existing Regular Expression:**

1. Locate the exiting RegExParseInt elements.

2. Modify the desired RegExParseInt element as required:

```xml
<RegExParseInt>((\W\d{1,3}\d{0,4}\s{0,5})*</RegExParseInt>
<RegExParseInt>(([+-])\d{0,6}\.*\d{0,6})*</RegExParseInt>
```

**WARNING!**
The RegExParseInt controls the way the Point data is parsed. Changing this file can break this converter. Changes to the RegExParseInt should only be made by a person who has a working knowledge of Regular Expressions.

### 4.3.5 Point File Conversion

1. If the Points to Robot Job Converter application is not already running, open it from the MotoSim Points Importer EG command menu.

2. Begin by loading your configuration files.
a. Click on the folder icon for the Cell configuration file.

b. Locate the file you want to use and click Open. If Points to Robot Job Converter displays the cell file in Red, there may be a problem with the selected file. Make certain you have selected the correct cell configuration file or edit the file to correct the problem.

c. Click on the folder icon for the Job configuration file.

d. Locate the file you want to use and click Open. If Points to Robot Job Converter displays the job file in Red, there may be a problem with the selected file. Make certain you have selected the correct cell configuration file or edit the file to correct the problem.

e. Click on the folder icon for the Points configuration file.

f. Locate the file you want to use and click Open. If Points to Robot Job Converter displays the cell file in Red, there may be a problem with the selected file. Make certain you have selected the correct cell configuration file or edit the file to correct the problem.

3. Click on the Points File button from the Converter Setup toolbar.

4. Locate the points file you want to convert and click Open. Points to Robot Job Converter begins processing the files you have selected to create a robot JBI file.

5. Upon success, you will receive a message stating that the points file has been successfully converted into a robot job. Click OK.

6. You may now play the converted job in MotoSim EG.
Notes
Chapter 5

G-Code to Robot Job Converter

The G-Code to Robot Job Converter application is a fully customizable tool for converting standard machine tool CNC G-Code programs into Motoman robot programs. G-Code Converter allows custom mapping of machine functions to robot functions for any G-Code format such as SURFCAM®, GibbsCAM® or other proprietary G-Code file formats. I/O and other commands can also be adjusted for CNC-to-robot variations.

G-Code Converter is ideally suited for customers who are working with third-party CAD/CAM expert packages for applications such as material removal, grinding, mold creation, surface finishing or drilling and tapping. It is a productivity enhancement tool for users that have a mix of CNC machines and robots in their shop, or individuals who know G-Code programming.

Features include:

- 3-axis and 5-axis machine code can be converted to process your 3D parts
- Convert existing machine tool operations into robot programs off-line
- Automatically splits large processes into multiple subroutines for robot operations
- Allows custom mapping of machine functions to robot functions
- Three application examples provided
- Customizable conversion configuration file

The figure below provides an overview of the G-Code Converter process including intermediate files that are produced.

The G-Code file is first converted into a master XML file. This file is then broken into master and callable subroutines XML job files. Each of these files has both the XML header section as well as the point location data. These XML files are now compatible with the main MotoSim EG Points Importer application.
The third and fourth conversions are triggered by the G-Code Converter application but are actually performed by the MotoSim Points Importer and the MotoSim EG applications. These conversions produce an intermediate set of INF files that are used by MotoSim EG to produce the final set of robot JBI job files.

5.1 Installing G-Code Converter

The G-Code to Robot Job Converter application is a productivity add-on to the MotoSim EG Points Importer client application and therefore requires both the MotoSim EG Points Importer (P/N 148611-2) and MotoSim EG (Full 151021-1 or Lite 151019-1). All MotoSoft software is provided on a single CD-ROM with a browser/installer utility (Motoman CD-ROM Browser, P/N 141720-1). Please refer to Section 1.5 for detailed installation instructions.

5.2 First Time Use

The G-Code to Robot Job Converter application is accessed from MotoSim EG Points Importer using the [G-Code to JBI] button on the toolbar.

Before you begin using MotoSim Points Importer EG, you must first have a valid MotoSim EG .mcl robot cell file and XML configuration file using the correct schema. The MotoSim EG cell must have a robot, TCP or Tool Center Point set, and have a user frame (part) coordinate system setup. After first time use, MotoSim EG Points Importer will automatically start MotoSim EG and open the last cell modified.

5.2.1 Sample Files

G-Code Converter includes several sample files that you can open and use to get started with G-Code Converter. These sample files are located in the Example Cells folder in the MotoSim EG Points Importer folder (c:\Program Files\Motoman\MotoSim EG Points Importer\Example Cells\).
5.3 Understanding the G-Code Converter Work Area

The G-Code Converter screen uses sequentially activated buttons to guide you through the conversion process. A status bar notifies you of where you are in the process. A series of tabbed windows display detailed information regarding the files used and also allow you to edit process files to achieve the most accurate conversion.

5.3.1 Command Menus

File
Contains the Load Configuration File and Convert CNC File process commands as listed below.

Robot
Contains the Play Robot Job(s) command.

Help
- About - Contains version information.
- Contents - Opens the interactive Help file.
### 5.3.2 Process Toolbar

The Process Toolbar contains the main operation buttons for G-Code Converter. The buttons become available from left to right as the process unfolds.

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Load Configuration</strong></td>
<td>The Load Configuration button allows you to browse to the desired configuration XML file. Once the configuration file has been selected, the [NC to JBI] and [NC to XML] buttons become available, and the config status bar turns green.</td>
</tr>
<tr>
<td><strong>NC to JBI</strong></td>
<td>The NC to JBI button converts the G-Codes directly to robot jobs in one step using only the settings in the XML configuration file.</td>
</tr>
<tr>
<td><strong>NC to XML</strong></td>
<td>The NC to XML button converts the G-Codes to an intermediate XML file. This allows you to edit and troubleshoot the XML file before the converting to robot jobs.</td>
</tr>
<tr>
<td><strong>XML to JBI</strong></td>
<td>The XML to JBI button completes the conversion process after you have converted the NC files to XML.</td>
</tr>
<tr>
<td><strong>Take 2</strong></td>
<td>The Take 2 button selects and plays the complete set of JBI files created from the G-Code.</td>
</tr>
<tr>
<td><strong>XML Edit</strong></td>
<td>The XML Edit button opens the XML configuration file in an editor allowing the user to edit the file.</td>
</tr>
<tr>
<td><strong>NC Edit</strong></td>
<td>The NC Edit button opens the G-Code file in an editor allowing the user to edit the file.</td>
</tr>
<tr>
<td><strong>Abort</strong></td>
<td>The Abort button cancels the G-Code conversion.</td>
</tr>
</tbody>
</table>

### 5.3.3 Status Bar

The Status bar displays the current status of the conversion process. Each window is color coded to show the status of each process file. Red indicates that there is a problem with this file. Yellow indicates that the file has yet to be loaded, and Green means that all files are correct and the application is ready to convert your G-Code file.
5.3.4 Tabbed Panels

5.3.4.1 XML Config File Panel

The XML Config File panel displays the XML configuration file.

Note: Editing is not possible directly from the XML Config File panel. Use the Edit XML button to open the XML configuration file in an editor.

To expand and contract the tree:

You can expand and contract the entire tree view using the contract [<>] and expand [<>] buttons located at the top of the panel. You can also expand and contract individual elements using the [+] and [-] symbols located next to each element.

The vertical and horizontal scroll bars are made available to view different portions of the expanded tree.
5.3.4.2 NC File Panel

The NC File panel provides information regarding the G-Code NC file. The G-Code File Status area provides information regarding the G-Code NC file including; file status and number of valid lines.

Specific lines can be found within the G-Code using the Find In NC portion of the panel. Simply enter the desired code in the search window and press [Find]. NC code lines are displayed in the right hand window. Pressing the [Goto in NC File] button highlights the lines in the NC file display window at the bottom portion of the panel.

The NC file code is displayed in the lower left window. This window can be configured to display specific types of code. Using the selection window, you can view either raw input lines, or only the lines, functions or positions handled during the conversion. Simply select the desired view and click the [Refresh G Code Display] button.
5.3.4.3 NC to XML Instructions Panel

The NC to XML Instructions panel displays the modified G-Code.

To expand and contract the tree:

You can expand and contract the entire tree view using the contract [><] and expand [<>] buttons located at the top of the panel. You can also expand and contract individual elements using the [+ ] and [-] symbols located next to each element.

Vertical and horizontal scroll bars are made available to view different portions of the expanded tree.

Blue highlighting indicates new lines that have been inserted into the code. Beige highlighting indicates code that has been changed or modified.

The Line Number Options portion of the panel allow the user to view the code as either Raw G-Code, or Handled - One Per Line.
5.3.4.4 Robot XML Files Panel

The Robot XML Files panel shows the robot XML files created from the G-Code.

To expand and contract the tree:

You can expand and contract the entire tree view using the contract [><] and expand [<>] buttons located at the top of the panel. You can also expand and contract individual elements using the [+ ] and [- ] symbols located next to each element.

Vertical and horizontal scroll bars are made available to view different portions of the expanded tree.
5.3.4.5 **Robot JBI Files Panel**

The Robot JBI Files panel shows the robot JBI files created from the intermediate Robot XML files.

![Robot JBI Files Panel](image)

**To expand and contract the tree:**

You can expand and contract the entire tree view using the contract [><] and expand [<>] buttons located at the top of the panel. You can also expand and contract individual elements using the [+] and [-] symbols located next to each element.

Vertical and horizontal scroll bars are made available to view different portions of the expanded tree.

5.3.4.6 **JBI Log Panel**

The JBI Log panel provides a list of JBI files created.

![JBI Log Panel](image)
5.3.4.7 Summary Panel

The Summary panel provides the user with a place to enter notes and comments regarding the G-Code conversion and create a complete status report.

The status report is saved as a PDF file and contains all information regarding the G-Code conversion including files used and created and any errors that occurred during the process.

5.3.4.8 Config. Control Panel

The Config. Control Panel allows the user to make temporary changes to the XML configuration file.
**Debug Options**
The Debug Options portion of the panel allows you to make changes to the Default Tool #, User Frame#, Length Units, and Time Units. You can also include line numbers as comments in the JBI file as well as disable warning message boxes.

**Default Orientation**
The Default Orientation frame allows quick, temporary changes to the Rx, Ry, and Rz for troubleshooting and conversion adjustments.

**CNC to Robot Orient. Mapping**
The CNC to Robot Orient. Mapping frame allows you to make quick, temporary changes to the orientation mapping for quick adjustments. Permanent changes must be made in the XML configuration file.

**Default Joint and Linear Velocities**
Default Joint and Linear velocities can also be temporarily adjusted from the Config. Control Panel.

### 5.3.5 Current CNC File Status Bar

The Current CNC File Status Bar lists the current CNC file and it’s status.

### 5.3.6 Current Config File Status Bar

The Current Config File Status Bar lists the current config file and it’s status.

### 5.3.7 Transparency

The transparency slider allows you to make the G-Code to Robot Job Converter application work area appear transparent. When this is done, other application windows can be seen through the work area. This can allow you to view a robot job running in the MotoSim EG application window while continuing work in the G-Code to Robot Job Converter application work area.

#### 5.3.7.1 On Top Check Box

The On Top check box allows you to set the G-Code to Robot Job Converter application to always appear on top of other tiled windows or applications.

### 5.4 Opening G-Code to Robot Job Converter

The G-Code to Robot Job Converter application is accessed directly from the MotoSim Points Importer EG command menu.
5.5 XML Configuration File

The XML configuration file is the heart of the G-Code Converter system. XML files are markup files that allow the user to create any markup tag needed. The XML configuration file tells MotoSim Points Importer EG how to translate an NC instruction into a single, or group of Motoman JBI instructions.

Two sample XML configuration files are provided with this software. These sample files are very broad in their application. You may find that they serve your needs without modification. However, if you find you need to make modifications, do not modify the original files. Keep these original sample files in a safe location. Should you need to make modifications to these files, make all changes to copies and not the original files.

The XML configuration file is made up of two major parts; the File Configuration, and specific G-code to Inform conversion directions.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!-- edited with XMLSpy v2004 rel. 4 U (http://www.xmlspy.com) by Jim Johnson (None) -->
<!-- <Root>     + <FileConfiguration>     + <GTOINFORM>     + <GTOINFORM>     + <GTOINFORM>   </Root> -->

5.5.1 File Configuration>

The <File Configuration> element contains generic instructions for handling the G-Code conversion. Required child elements include:

- RegExParseInt
- ParseGroups
- InformFileFormats
- AngleSystem
- MotionFunctions
- Units
- ThreeDOF
- ReferenceModel
- ArmConfig
- AlignXwithPath
- CircleFixUp
- CircleIJCenter
Several optional elements are also used depending on the Degrees of Freedom (DOF) and other specifics of the G-Code. The following is a sample of a File Configuration element:

- <FileConfiguration>
  <RegExParseInt>([T]\d{1,3})?([GM]\d{1,3})?([D]-[^d]{0,3}\.*d{0,4})? ([X]-[^d]{0,3}\.*d{0,4})?([Y]-[^d]{0,3}\.*d{0,4})?([Z]-[^d]{0,3}\.*d{0,4})?([I]-[^d]{0,3}\.*d{0,4})?([J]-[^d]{0,3}\.*d{0,4})?([A]-[^d]{0,3}\.*d{0,4})?([B]-[^d]{0,3}\.*d{0,4})?([F]-[^d]{0,3}\.*d{0,4})?\</RegExParseInt>
  <ParseGroups G1="T" G2="GM" G3="D" G4="X" G5="Y" G6="Z" G7="I" G8="J" G9="A" G10="B" G11="F" />
  <ThrowAwayLines>(</ThrowAwayLines>
  <ThrowAwayLinesRegEx>([N]\d{1,4})\)\</ThrowAwayLinesRegEx>
  <ParseGroupsThrowAwayLines G1="N" G2="(" />
  <StartFile>%</StartFile>
  <InformFileFormats MasterandSubs="True" MasterName="gmaster" MaxLines="750" />
  <Units>
    <LengthUnits>Millimeters</LengthUnits>
    <AngleUnits>Degrees</AngleUnits>
    <TimeUnits>Seconds</TimeUnits>
  </Units>
  <ThreeDOF Rx="180.0" Ry="0.0" Rz="0" />
  <ReferenceModel JobTypeMaster="True">UF1</ReferenceModel>
  <ReferenceModel JobTypeSub="True">UF1</ReferenceModel>
  <ArmConfig JobName="default">
    <S>-34160</S>
    <L>-18343</L>
    <U>-32472</U>
    <R>0</R>
    <B>8492</B>
    <T>9125</T>
  </ArmConfig>
</FileConfiguration>
5.5.1.1 RegExParseInt

The RegExParseInt element is a required element. It is a Regular Expression instruction. This instruction tells the G-Code Converter which G-Code Instructions to look for and the format of the G-Code file.

The instruction is written so that most G-Code files can be converted without modifying this element. Many G-Code fields are optional and may or may not appear in the G-Code file. For example, line Numbers may or may not be included in the G Code instruction.

Example:

```xml
<RegExParseInt>(N\d{1,5})+\s*([MG]\d{1,3})\*\s*([D][-]*\d{0,3}\.*\d{0,4})\*\s*([X][-]*\d{0,3}\.*\d{0,4})\*\s*([Y][-]*\d{0,3}\.*\d{0,4})\*\s*([Z][-]*\d{0,3}\.*\d{0,4})\*\s*([I][-]*\d{0,3}\.*\d{0,4})\*\s*([J][-]*\d{0,3}\.*\d{0,4})\*\s*([A][-]*\d{0,3}\.*\d{0,4})\*\s*([B][-]*\d{0,3}\.*\d{0,4})\*\s*([F][-]*\d{0,3}\.*\d{0,4})\*</RegExParseInt>
```

It can be broken down into groups of instructions. The line number parsing instruction is shown below:

```xml
(N\d{1,5})+\s
```

This instruction says to look for an N followed by 1 to 5 numerical digits. The + indicates that the value must occur 1 or more times to match, and the \s says that the NXXXX may be followed by a white space character, a tab, form-feed, or carriage return.

The M or G command is parsed by the follow RegEx expression:

```xml
([MG]\d{1,3})\*\s*
```

This expression say to look for an M or a G followed by 1 to 3 numerical digits. The * says that match may occur 0 or more times. The \s* indicates that white spaces may or may not be included after the M or G instruction.

The X location instruction is parsed with the following command;

```xml
([X][-]*\d{0,3}\.*\d{0,4})\*\s*
```

This instruction is similar to the ones above except that the X may or may not include a - or negative sign, 0 to 3 digits, or period may or may not be included followed by 0 to 4 additional digits.

More information is available on the Web at http://regex.oshervov.com

⚠️ **WARNING!**

Corruption of the RegExParseInt element may disable G-Code Converter. Always make a backup of the original instruction before making any modifications.
5.5.1.2 ParseGroups

The ParseGroups element is a required element. This instruction works with the RegExParseInt instruction. It identifies each group command. If a group is dropped from the RegExParseInt, it must be taken out of this command. The order of the commands must also match the order of the RegExParseInt commands.

Example:

    <ParseGroups G1="N" G2="GM" G3="L" G4="F" G5="D" G6="X" G7="Y" G8="Z"
                 G9="I" G10="J" G11="K" G12="A" G13="B" G14="C" G15="R"
                 G16="H" G17="F"/>

5.5.1.3 StartFile

The StartFile element is an optional element that defines the special character used in the G-Code to determine the start of the code. Default value is %.

All information above the start file command will be ignored. If this command is not present, the G code Converter will start looking for valid commands immediately.

Example:

    <StartFile>%</StartFile>

5.5.1.4 ThrowAwayLines

The ThrowAwayLines element is an optional element that allows you to define lines of G-Code that you would like to have ignored during the conversion process.

5.5.1.5 ThrowAwayLinesRegEx

The ThrowAwayLinesRegEx element is an optional element. This is the same as the ThrowAwayLines command but more than one character can be included. It is also a Regular Expression and can be used to catch more difficult sequences of characters that signify the start of a comment line.

5.5.1.6 ParseGroupsThrowAwayLines

The ParseGroupsThrowAwayLines element is a required element. This instruction works with the ThrowAwayLinesRegEx instruction. It identifies each group command. If a group is dropped from the RegExParseInt, it must be taken out of this command. The order of the commands must also match the order of the RegExParseInt commands.

Example:

    <ParseGroupsThrowAwayLines> G1="/"</ParseGroupsThrowAwayLines>

5.5.1.7 InformFileFormats

The InformFileFormats element is a required element.

- **MasterName**: obsolete - left in backwards compatibility - do not modify
- **MaxLines**: Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.
Example:

```xml
<InformFileFormats MasterandSubs="False" MasterName="gmaster"
MaxLines="900" CallSub="M10"/>
```

**CallSub**

Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

Example:

```xml
<InformFileFormats MasterandSubs="False" MasterName="gmaster"
MaxLines="750" CallSub="M10"/>
```

**ReturnSub**

Creates a RET statement in the current Robot Job when a specific M or G code is found.

Example:

```xml
<InformFileFormats MasterandSubs="False" MasterName="gmaster"
MaxLines="750" CallSub="M11"/>
```

5.5.1.8 **AngleSystem**

The AngleSystem element is a required element that defines the reference system for position orientation angles used in the cell. Values are either YPR (Yaw, Pitch, and Roll) or Euler. YPR is the Motoman and MotoSim EGEg standard.

Euler angles is a method of defining three rotations of an object about a point. Euler angles are used in the Aerospace and Graphics industry Euler Angles generally rotate about the ZYZ axes.

Example:

```xml
.AngleSystem>YPR</AngleSystem>
```

5.5.1.9 **MotionFunctions**

The MotionFunctions element is a required element. Lists of all G code Functions that map into robot moves. This is used as an aid to properly converting G Code files. Each G Code Function is listed in a MotionFunction Command which is shown below.

5.5.1.10 **FourToSixAxis**

This group of commands defines how the CNC A, B and C tool rotational axes are mapped into Robot Tool Angles Tx, Ty, Tz.

Normally the A CNC Angle is a Tool X rotation, the B CNC Angle rotates the Tool around the CNC Tool Y Axis and the C CNC Angle rotates the CNC about the CNC Tool Z axis.

The Z1, Y, Z2 Attributes come form the Euler angle naming convention. You can substitute or think of the Z1 as Tx, Y as Ty and Z2 as Tz robot angles.

For three degree of Freedom applications (X, Y, Z with a fixed tool orientation) this section may be omitted from the Configuration File.

The following sub tags or commands `<Z1Negate>`, `<YNegate>`, and `<Z2Negate>` tell the robot if the A, B, and C angles should be negated when converting them to Tx, Ty, or Tz.

Example:

```xml
<FourToSixAxis Z1="A" Y="B" Z2="C"/>
```
5.5.1.11 Units

The Units element is a required element that determines the units of measure for length, angle, and time. The LengthUnits, AngleUnits, and TimeUnits child elements define these values.

5.5.1.12 ThreeDOF

The ThreeDOF element is a required element that uses the Rx, Ry, and Rz attributes to define the tool orientation. This is the initial robot orientation (Tx, Ty, Tz) required to place the last length of the robot Tool in a configuration that makes the tool perpendicular to the User Frame or GCode part coordinate system.

CNC A = 0, B=0, C=0 makes the CNC tool perpendicular to the same robot User Frame or G Code part coordinate system.

This is a required element that is used in all G Code conversions (3 to 6 degree of freedom) applications.

Example:

```xml
<ThreeDOF Rx="180" Ry="0" Rz="180"/>
```

5.5.1.13 ReferenceModel

The ReferenceModel element is a required element that uses the JobTypeMaster, and JobTypeSub attributes.

5.5.1.14 ArmConfig

The ArmConfig element is a required element that uses the S,L,U,R,B, and T child elements to define the initial robot arm starting position and form. This position is specified in joint coordinates and sets the initial pose (elbow up/dow, etc.) of the robot. This position can be set by moving the simulated robot to the part coordinate system 0,0,0 location and setting the individual axes to archive the desired pose. The values can be copied from MotoSim EG and pasted in the appropriate joint location.

Note: You can specify up to 21 unique robot configurations. The first set of Arm Configuration values are automatically selected. Other Configurations can be selected by the SetPulseConfig commands.

5.5.1.15 AlignXwithPath

The AlignXwithPath element is an optional element that is reserved for future use.

5.5.1.16 CircleFixUp

The CircleFixUp element is a required element that determines if G-Code Converter will analyze and insert additional circle points. Set this function to True for normal operation. This value may be temporarily set to False to help troubleshoot G-Code to Robot Conversion errors.
Example Use: `<CircleFixUp>True</CircleFixUp>`

5.5.1.17 CircleIJCenter
The CircleIJCenter element is a required element.

5.5.1.18 DefaultToolNumber
The DefaultToolNumber element is an optional element.

5.5.1.19 DefaultNCPath
The DefaultNCPath element is an optional element.

5.5.1.20 DCISimpleDrip
The DCISimpleDrip element is an optional element used to create a master job that will automatically download a subroutine while simultaneously executing another robot job or subroutine.

This option is usually used in Mold Creation applications specifically or in any application that creates large jobs. Some Mold jobs may exceed 1,000,000 program points. Since the robot does not have that type of local storage capacity, the majority of jobs must be located on a PC file server and downloaded, executed and then deleted to make room for other robot jobs.

This requires the Motoman Inc., Visual DCI communications software application to perform the on-line robot job downloads into the controller.

5.5.1.21 DCISimpleDripDVar
The DCISimpleDripDVar element is an optional element.

5.5.1.22 CannedVelocity
The CannedVelocity element is an optional element.

5.5.1.23 ExternalAxes
The ExternalAxes element is an optional element that uses the `<AxesPulsePosition>` and `<AxesType>` attributes to define external axes. This element is set to True when the Robot is equipped with an external Axis. Otherwise it is set to False. When `<ExternalAxes>` is set to True, this element stores the static external Axis pulse count. Current Converter only supports a static external axis.

5.5.2 `<GTOINFORM>`

`<GTOINFORM>` is a complex type element that uses the `GCODE` and `INFORM` child elements to define the G-Code to Inform conversion. For example, the following `<GTOINFORM>` element converts the G-Code function G00 (rapid position) to it’s Inform equivalent joint motion.

- `<GTOINFORM>`
  - `<GCODE func="G00">`
    `<Header>False</Header>`
5.5.2.1 <GCODE>

The <GCODE> element defines the G-Code. The `func` attribute names the actual G-code being converted. Eight child elements help define the G-code. The Header, Description, MotionFunction, and isLiteralInform elements are required, while the ShiftOffset, MotionGroup, EndCannedCycle StartCannedCycle elements are optional.
<table>
<thead>
<tr>
<th>Description</th>
<th>The Description element is a required element that is used in the GCode to Inform conversion process. It is only for G Code Converter Operator use. It does not affect the conversion of G-Code to Inform Instructions.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MotionFunction</td>
<td>The MotionFunction element is a required element.</td>
</tr>
<tr>
<td>isLiteralInform</td>
<td>The isLiteralInform element is a required element.</td>
</tr>
<tr>
<td>ShiftOffset</td>
<td>The ShiftOffset element is an optional element. True, False (default)</td>
</tr>
<tr>
<td>MotionGroup</td>
<td>The MotionGroup element is an optional element. True, False (default)</td>
</tr>
<tr>
<td>EndCannedCycle</td>
<td>The EndCannedCycle element is an optional element. True, False (default)</td>
</tr>
<tr>
<td>StartCannedCycle</td>
<td>The StartCannedCycle element is an optional element. True, False (default)</td>
</tr>
</tbody>
</table>

### 5.5.2.2 `<INFORM>`

The `<INFORM>` element is a complex element that uses several child elements to define the robot Inform code.
The MotionType element is an optional element that defines the motion type as either Circular, Joint, or Linear.

The OutputOn element is an optional element that is used to turn on controller outputs (i.e. OutputOn 3. XML: to JBI Instruction. Creates an Inform Instruction: DOUT OT(#X) ON.

The Timer element is an optional element that maps the robot controller timer.

The ThreeDOF element is an optional element. Rx, Ry, Rz (attributes)

The LengthUnits element is an optional element. Inches, Millimeters
The ReferenceModel element is an optional element. UserFrame 1-10 (UF1-UF10)

The OutputOff element is an optional element used to turn OFF controller outputs turned ON using the OutputOn element. XML to JBI Instruction. Creates a Inform Instruction: DOUT OT(#X) OFF.

The GenInform element is an optional element used during the G-Code to Robot Job or JBI Conversion process. The files are first converted to a set of intermediate XML Files. These XML Intermediate Files are the same files that are used in the Points Importer Conversion process.

The GenInform command is an abbreviation for General Inform. It is used when there is not a specific XML To Robot JBI Command.

This is a very powerful command that allows the user to create any Inform Command and have it passed through to the Robot Conversion interface.

The CallJob element is an XML to Inform (.JBI) instruction. This element causes the robot to call a job subroutine.

This instruction appears in the intermediate XML file that the G Code Converter creates during the conversion process. It produces an Inform Instruction in the robot jbi.

Example:

```
<CallJob>Pickup</CallJob>
```

Example output JBI: CALL JOB:Pickup

Creates an inform command with the form: CALL JOB:xxx where xxx is the job name.

The SetPosElmnt element is the XML to JBI command that allow the user to set an element of a P Variable. There are two forms for this command.

The ShiftOn element is the XML to JBI command that allow the user to...
### 5.6 Getting Started (Quick Start)

The following section guides you through a sample conversion. This process can be used as a starting point for setting up your own configuration.

```
START
Open MotoSim EG Points Importer and Simulation Cell
Open Configuration File
Convert G-Code File to Robot JBI
Conversion Completed Without Error?
YES
NO
Modify G-Code File
YES
Modify G-Code File
NO
Solve Robot Issues
YES
Robot Errors?
YES
Create Conversion Report?
YES
Create Report
END
```
5.6.1 MotoSim EG Cell Setup

Before you can convert any G-Code, you must have a MotoSim EG cell file created. This cell must include a robot, Tool Center Point (TCP) or TCP set, and a user frame (part) coordinate system setup.

To create a new MotoSim EG Cell:

1. From the MotoSim EG main window, select File > New Cel Project... The New Cel window appears.

2. Enter the desired name for the cell in the File Name window and click Open. The new cell is created in MotoSim EG.

To add a robot to the cell file:

1. With the new cell open in MotoSim EG, select Robot > Add Robot... The Open window appears.

2. Browse to the desired robot folder and select the All.prm file and click Open. The Install Robot window appears.

3. If this is the correct robot, click OK. The robot model appears in the cell.

To add TCP offsets:

1. With the cell open and robot model visible, zoom into the T-axis of the robot.

2. Select Robot > Data Setting > Tool Data. The TOOL Editor window appears.

3. Modify the X, Z, and Ry values to reflect your tool.
To add Tool Model (Optional):

An optional tool model can be added to the cell to act as a visual aid for the TCP location.

1. From the MotoSim EG tool bar, select the CAD Tree button. The CAD Tree window appears.
2. With the tree expanded, right click on the robot tcp and select New Model... from the dropdown window. The Add Model Dialog window appears.
3. Enter the name of the tool model and click OK.
4. Click OK again to confirm file creation. The tool model appears in the CAD tree.
5. Double click on the new tool model file in the CAD tree. The tool model edit window appears.
6. From the Add Parts dropdown window, select CONE and click Add. The CONE Edit window appears.
7. Modify the L. Diameter to reduce the diameter of the cone and click OK.

8. Close the Tool Model window.

9. From the CAD Tree, right click on the tool model and select Select Position... from the dropdown window. The Position window appears.

10. Change the Z(mm) value to -100 to place the tip of the cone at the tool frame origin and click OK.

11. Close the CAD Tree view.

**To create a User Frame Model (Optional):**

1. From the MotoSim EG tool bar, select Robot > Reach View > Reach View. The Reach Area window appears.

2. Select the Tool radio button from the Mode panel.

3. Click Execute.

4. Click OK.

5. With the CAD tree expanded, right click on world and select New Model... from the dropdown window. The Add Model Dialog window appears.

6. Enter the name of the user frame model and click OK.

7. Click OK again to confirm file creation. The user frame model appears in the CAD tree.

8. Double click on the new user frame model file in the CAD tree. The user frame model edit window appears.
9. From the Add Parts dropdown window, select BOX and click Add. The BOX Edit window appears.
10. Modify the width and height to reduce the size of the box and click OK.
11. Close the User Frame Model window.
12. From the CAD Tree, right click on the user frame model and select Select Position... from the dropdown window. The Position window appears.

13. Change the X (mm) and Z(mm) values to place the user frame model in front of the robot model and within the robot reach.
14. Click OK.
15. Close the CAD Tree view.
16. From the MotoSim EG tool bar, select Robot > Reach View > Delete Reach View.

**To define the user frame:**
1. From the MotoSim EG tool bar, select Robot > Data Setting > Set User Frame... The Set User Frame window appears.

2. Using the cursor, select the corner of the user frame model.
3. De-select the Active radio button and click Close.

**To save the cell file:**
1. From the MotoSim EG tool bar, select File > Save.
5.6.2 XML Configuration File Setup

The XML Configuration File is the heart of the G Code Translation System. XML files are markup files that allow you to create any markup tags needed. This file tells the G code Converter how to translate an NC Instruction into a single or group of Motoman JBI Instructions.

The XML configuration file uses two lines to determine how the G code file is parsed.

To create an XML Configuration File from an Existing File:
1. Open the MotoSim EG cell folder you created in Section 5.6.1.
2. Create two new folders in the MotoSim EG cell folder. Name one folder “NC or GCode Files” and the other “XML Configuration Files.”
3. Place a copy of a sample XML configuration file into the XML Configuration Files folder.

To set units in XML configurations file:

The measurement units used in the XML Configuration file must match those in the G-Code file.
1. Open the XML Configuration file in an XML editor or text editor such as Notepad.
2. Locate the Units element.
   - <Units>
     - <LengthUnits>Millimeters</LengthUnits>
     - <AngleUnits>Degrees</AngleUnits>
     - <TimeUnits>Minutes</TimeUnits>
   - </Units>
3. Change the LengthUnits element to either Millimeters or Inches.
4. Change the AngleUnits element to either Degrees or Radians.
5. Change the TimeUnits element to either Minutes or Seconds.

To setup user frame in XML configuration file:
1. Locate the ReferenceModel element and record the User Frame (UF1-UF10).
   <ReferenceModel>UF1</ReferenceModel>

To setup initial arm configuration in XML configuration file:
1. From the MotoSim EG main window, click on the Position and Job View button from the Main tool bar. The Position and Job windows appear.
2. From the Position window, click on the Frame button.
3. Select the Pulse radio button from the Display Reference Frame window and click OK.

4. Locate the ArmConfig element in the XML Configuration file.

   <ArmConfig>
     <S>-6733</S>
     <L>118307</L>
     <U>87216</U>
     <R>-20336</R>
     <B>13994</B>
     <T>10689</T>
   </ArmConfig>

5. Copy and Paste the pulse position data for each of the axes (SLURBT) into the S, L, U, R, B and T child elements.

**To setup initial tool angles in XML configuration file:**

1. From the MotoSim EG main window, click on the Position and Job View button from the Main tool bar. The Position and Job windows appear.
2. From the Position window, click on the Frame button.
3. Select the User Frame UF# radio button from the Display Reference Frame window.
4. Enter the User Frame and click OK.
5. Set X, Y, and Z values to 0.
6. Set Rx, Ry, and Rz values.
To map CNC A, B, and C angles into robot angles:

To convert CNC A, B, and C Angles into Robot coordinates, they must be mapped to robot coordinates.

MotoSim Points Importer EG assumes that the CNC A, B, and C axes are rotating about the X, Y, and Z axes of the CNC Tool Frame. MotoSim Points Importer EG uses the intermediate User Frame or Part Coordinate system. This coordinate system is already being used to transform the CNC X, Y, and Z coordinates. Basically, G-Code Converter aligns the A, B, and C rotations of the CNC with the Rx, Ry, and Rz rotations in the user coordinate system.

There are several ways to perform this transformation. Four of the most common are shown below. In Example 1, CNC A (Tool Rx) angles are mapped into negative User Frame Rx angles. The negative occurs whenever the Tool axis is pointing in the direction opposite the corresponding User Frame or Part axis.

Examples 1 and 2 are preferred because they keep the A and Rx mapping in place.

Users must make sure they update the XML Configuration File once mapping is complete. Both the <FourToSixAxis> tag, and the <ThreeDOF> tag must be updated.

A, B, and C angles are translated into the appropriate robot tool angles when the *.ncc file is converted into robot jobs or *.jbi files.

Note: If reach problems exist, these values may need to be adjusted.

Note: Various combinations can be tried temporarily by updating the <FourToSixAxis> Tag and the <ThreeDOF> Tag

Example 1:

In this example, the X-axis is the same for both the tool and user frame, while the Y, and Z-axes are opposite or negated.
XML Configuration Excerpt For Example 1

```xml
<FourToSixAxis Z1="A" Y="B" Z2="C">
  <Z1Negate>False</Z1Negate>
  <YNegate>True</YNegate>
  <Z2Negate>True</Z2Negate>
</FourToSixAxis>
<ThreeDOF Rx="180.0" Ry="0.0" Rz="0"/>
```

**Example 2:**

In this example, the Y-axis is the same for both the tool and user frame, while the X, and Z-axes are opposite or negated.

```
<FourToSixAxis Z1="A" Y="B" Z2="C">
  <Z1Negate>True</Z1Negate>
  <YNegate>False</YNegate>
  <Z2Negate>True</Z2Negate>
</FourToSixAxis>
<ThreeDOF Rx="180.0" Ry="0.0" Rz="0"/>
```

<table>
<thead>
<tr>
<th>RxTool = RxPart</th>
<th>RyTool = -RyPart</th>
<th>RzTool = -RzPart</th>
</tr>
</thead>
<tbody>
<tr>
<td>A=Rx</td>
<td>B=-Ry</td>
<td>C=-Rz</td>
</tr>
</tbody>
</table>

XML Configuration Excerpt For Case 2

```xml
<FourToSixAxis Z1="A" Y="B" Z2="C">
  <Z1Negate>True</Z1Negate>
  <YNegate>False</YNegate>
  <Z2Negate>True</Z2Negate>
</FourToSixAxis>

<ThreeDOF Rx="180.0" Ry="0.0" Rz="180"/>
```

<table>
<thead>
<tr>
<th>RxTool = -RxPart</th>
<th>RyTool = -RyPart</th>
<th>RzTool = -RzPart</th>
</tr>
</thead>
<tbody>
<tr>
<td>A=-Rx</td>
<td>B=Ry</td>
<td>C=-Rz</td>
</tr>
</tbody>
</table>
Example 3:
In this example, the Z-axis is negated, while the X, and Y-axes are 90 degrees off.

<table>
<thead>
<tr>
<th>RxTool = RyPart</th>
<th>RyTool = -RxPart</th>
<th>RzTool = -RzPart</th>
</tr>
</thead>
<tbody>
<tr>
<td>A=Ry</td>
<td>B=Rx</td>
<td>C=-Rz</td>
</tr>
</tbody>
</table>

XML Configuration Excerpt For Case 3

```xml
<FourToSixAxis Z1="B" Y="A" Z2="C">
    <Z1Negate>False</Z1Negate>
    <YNegate>False</YNegate>
    <Z2Negate>True</Z2Negate>
</FourToSixAxis>

<ThreeDOF Rx="180.0" Ry="0.0" Rz="90"/>
```
Example 4:
In this example, the Z-axis is negated, while the X, and Y-axes are -90 degrees off.

XML Configuration Excerpt For Case 4

```xml
<FourToSixAxis Z1="B" Y="A" Z2="C">
  <Z1Negate>True</Z1Negate>
  <YNegate>True</YNegate>
  <Z2Negate>True</Z2Negate>
</FourToSixAxis>

<ThreeDOF Rx="180.0" Ry="0.0" Rz="-90"/>
```

To setup tool:
1. Before proceeding, verify that the TCP is properly setup.
2. Set Tool X, Y, and Z values to 0 in the selected user frame.
3. Change Orientation so the tool (last link) is perpendicular to the user frame.
4. Use Tool Rz to rotate the tool so that the Tool Rx and the User Frame Rx are aligned (Example 1 above) or one of the other configurations shown in Examples 2, 3, or 4.
5. Copy the Tool Rx, Ry, and Rz values (in User Frame coordinates and place them in the XML Config File in the ThreeDOF Tag.
<ThreeDOF Rx="180.0" Ry="0.0" Rz="180"/>

   a. Determine which example best represents your tool setup (example 1, 2, 3 or 4 above).
   b. Set the correct values in the XML Configuration File.

```xml
<FourToSixAxis Z1="B" Y="A" Z2="C">
    <Z1Negate>True</Z1Negate>
    <YNegate>True</YNegate>
    <Z2Negate>True</Z2Negate>
</FourToSixAxis>
```

### 5.6.3 Standard CNC File Conversion

1. Press the Load Configuration button on the GCode Converter toolbar and browse to the desired XML Configuration file.

2. Click on the NC to JBI button to convert the GCode file directly to JBI.
   OR

3. Click on the NC to XML button to first convert to XML. This allows you to modify the files before converting.

4. Click on the XML to JBI button to finish the complete process.

### 5.6.4 Circles in CNC and Robots

#### 5.6.4.1 Circles in CNCs

G02: Circular motion, clockwise direction. There are two circular motion commands for move the machine tool in arcs, which are segments of circles, and full circles.

The G02 command is the instruction for a clockwise circular motion. The arc starts at the tool's current location and sweeps around clockwise to the designated ending location at the programmed velocity.
There are several 'standard' methods to define the radius of the arc. These methods are discussed below, after the discussion on the counter-clockwise command.

G03: Circular motion, counter-clockwise direction. The G03 command is the instruction for a counter-clockwise circular motion. The arc starts at the tool’s current location and sweeps around counter-clockwise to the designated ending location at the programmed velocity.

There are several 'standard' methods to define the radius of an arc. These methods are discussed below, after the discussion on the counter-clockwise command.

**Circular motion format, Arc Radius Definitions**

**Circles:**
All standard circular motion G-code instructions program the arc's radius. However, some CNC controls support only one format, while others support two or more formats. Refer to your machine's programming manual to determine which formats your CNC control supports.

The most commonly supported format uses I and J words to define the arc's center point location. Unfortunately, there are several different formats for defining the I and J locations!

One format that is pretty straightforward, programs the arc center-point using absolute coordinates. The arc's center-point location in the X axis is programmed with the 'I' word and the arc's center-point location in the Y axis is programmed with the 'J' word. The following example uses this format to make a 0.5 inch radius:

```
N108 G1 X10.596 Y4.7077 Z-1.875 C0. B0. F500.  -- --- Current location
N110 G2 X10.596 Y4.7077 Z-1.875 C0. B0. I-1.2729 J0. K0. --- End of Circle Arc
```

**5.6.4.2 Circles in Robots**

To program circles with the robot:

Robots define circular paths by programming 3 or more points on the circle parameter. There is no Radius information of any kind in the robot job.
Because of these differences, the G-Code Converter must add additional Circle points to the G-Code Data. The following picture shows the new and modified points that have been added to the G-Code position data.

The Blue shaded points are new points that have been added to the G Code Data. The brown highlighted point has been modified.

This is controlled by two elements in the XML Configuration file. These elements are the <CircleFixup> Command and the <CircleIJCenter> Command. When the CircleFixUp command is True, then the additional Circle data is inserted in the robot job. If CircleFixUp command is False, then the circle processing is skipped.

CircleIJCenter is True when the Radius I J and/or K values shown in the circle command gives the circle radius coordinates relative to the center of the circle. When the CircleIJCenter is False, the I, J, K values are defined as absolute part coordinates. CNC Programs may provide either of these types of radius coordinates.
To modify G-Code comments:

The Debug Options portion of the Config. Control panel allows you to include line numbers as comments in the JBI file as well as disable warning message boxes. This is very helpful when debugging the conversion by allowing you to see exactly which line of the job may be causing problems.

5.6.5 Rotating Station

G-Code Converter can convert CNC Jobs that have the part on a rotating axis. See G-Code Example 4.cel.

The Rotational axis transformation is controlled by the <RotatingStationSetup> Commands in the G-Code Configuration File. Most of these values are self explanatory with the exception of the PartCordinateSystemRotates with Part command which is explained below.

PartCordinateSystemRotateswithPart = True or False. When false, the part coordinate system does not rotate with the part (i.e. Z axis is up regardless of the rotational axis value). This setting is generally used with symmetric parts such as circular or round parts.
<RotatingStationSetup>
  <RotationStationName>ST1</RotationStationName>
  <RotatingStationUserFrameNumber>1</RotatingStationUserFrameNumber>
  <PartCordinatesRotateWithStation>False</PartCordinatesRotateWithStation>
  <DefaultRotatingVelocity>10</DefaultRotatingVelocity>
  <DefaultRotatingMotionType>Joint</DefaultRotatingMotionType>
  <DefaultRotatingUnits>Degrees</DefaultRotatingUnits>
  <RotatingStationToCNCAngleMapping>
    <StationtoCNCAxis>
      <StationAxisNumber>1</StationAxisNumber>
      <CNCAxisName>A</CNCAxisName>
      <NegageCNCAxis>False</NegageCNCAxis>
    </StationtoCNCAxis>
  </RotatingStationToCNCAngleMapping>
</RotatingStationSetup>

In the corresponding G-Code File, the A value is the value, in degrees, of the rotational station axis.

To transform G-Code jobs with parts on a rotational axis:
1. Setup MotoSimEG cell.
   a. Add a rotational axis model to the MotoSimEG Cel file.
   b. Load an All.prm file from a robot that has a 7 rotational axis defined. This all.prm file must be placed in the robot subdirectory of the MotoSimEG Cel file.
   c. Create a User Frame at the center of the rotational axis.
2. Add the "RotatingStationSetup" commands to the G-Code Configuration Cell.
3. Open and transform G-Code files with one axis defined as a rotational axis.
5.6.6 Robot on a Rail

For large volume parts, the robot may be added to a rail. See G-Code Example 5.cel for an operational example of this concept.

To setup a Robot on a Rail:

1. Setup MotoSimEG Cell.
   a. Add a Linear axis model to the MotoSimEG Cel file.
   b. Load an All.prm file from a robot that has a 7 Linear axis defined. This all.prm file must be place in the robot subdirectory of the MotoSimEG Cel file.
   c. Create at least one User Frame for Static or Dynamic rail as defined in the following sections.

2. Add the "Static or Dynamic Rail Option" commands to the G-Code Configuration Cell.

3. Open and convert G-Code files

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. Note that 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 (for example, long cutting or de-burring long strips of material).
Note: G-Code jobs can be divided into subroutines in which some work as a Static Rail and some act as a Dynamic Rail system.

5.6.6.1 Robot on a Rail Setup - Static Rail Option

Add the following command to the G Code Configuration file.

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

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.

5.6.6.2 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.

Note: The robot axis that is aligned with the rail will become static (i.e. in this example the robot would maintain a value of $X = 700.0$ mm and offsets from this value will be added to 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 could be increased to extend the robot arm and thus lower the robot elbow. It might also be made smaller to bring the tool closer to the robot base thereby making the process run more smoothly.

```xml
<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>
```
5.6.6.3 **XYZ or Relative Job Output Option**

To make the output job a relative job (Cartesian coordinates), add the following commands to the G-Code configuration file.

All RobotPosture commands should be set to 0. MotoSimEG will automatically assign them the correct values.

```xml
<JobOutputOptions>
  <JobType>
    <RectanJobType>
      <PositionConfiguration>
        <UserFrame>1</UserFrame>
      </PositionConfiguration>
      <RobotPosture>
        <FlipPosture>0</FlipPosture>
        <UpperPosture>0</UpperPosture>
        <FrontPosture>0</FrontPosture>
        <RPosture>0</RPosture>
        <TPosture>0</TPosture>
      </RobotPosture>
    </RectanJobType>
  </JobType>
</JobOutputOptions>
```

5.6.6.4 **XYZ or Pulse Job Output Option**

Job outputs can produce Pulse type jobs if the following command is added to the G-Code Configuration file, or if the JobOutputOptions command is completely removed from the G-Code Configuration File.

```xml
<JobOutputOptions>
  <JobType><PulseJobType>True</PulseJobType></JobType>
</JobOutputOptions>
```
5.6.7  **MotoCal Calibration Tasks For Offline Filter Creation**

Jobs created in MotoSim EG are based on an ideal Robot.

**MotoSim EG Simulation Tasks Part 1:**
1. Create a MotoSim EG cell file.
2. Add Robot to the simulation cell.
3. Create a CAD model of the Workplaces, Holes and Circles - Optional
4. Place work plate Model in the Simulation Cell
5. Created an ideal user frame who's origin is at the first hole

5.7  **Troubleshooting G-Code to Robot Conversions**

Occasionally, errors occur during the conversion. These errors occur most frequently as a new cell is being setup. By far the most common errors are robot reach errors (Conversion Error # 9). These errors can be caused by a number of sources including: initial Robot Pose or starting position, poor part placement (user Frame positioning errors), bad tool mounting, bad CNC ABC angle to Robot Tool mapping, incorrectly sized robot (need a bigger robot).

The key to troubleshooting robot reach errors is to look for multiple indications.

The following is a list of possible errors and their solution.
### Table 5  G Code Conversion Errors and Possible Solutions

<table>
<thead>
<tr>
<th>General Category</th>
<th>Indication</th>
<th>Error Explanation</th>
<th>Possible Solution</th>
</tr>
</thead>
</table>
| Reach Error      | - Model to Large  
- CreateJBJIErr=9 | Configuration file Linear Units are set at millimeters and should be Inches | - Change Configuration file `<LengthUnits>` to inches `<Units>`  
 `<LengthUnits>Inches</LengthUnits>` `<AngleUnits>Degrees</AngleUnits>` `<TimeUnits>Minutes</TimeUnits>` `</Units>` |
|                  | - CreateJBJIErr = 9  
- Some Points are out of reach of the robot | Robot Reach Error | - Adjust the location of the user frame and try the conversion again  
- Try changing the Robot Tool TCP |
|                  | - CreateJBJIErr = 9  
Error_txt file has robot limits | Robot Reach Error - One axis is reaching a limit | - Adjust location of user frame and try conversion again  
- Try changing Robot Tool TCP  
- Change robot's Initial pose or start |
|                  | - CreateJBJIErr = 9  
- Points occur at wrong location | Wrong user frame selected. In cells with more than one user frame, it is possible to select the wrong user frame. | - Open the Configuration File and change the User Frame to the correct User Frame number `<ReferenceModel JobTypeMaster="True">UF6</ReferenceModel>` `<ReferenceModel JobTypeSub="True">UF6</ReferenceModel>` |
| Setup Error      | Error Code=8  
Create Model Error | User Frame specified in Configuration file does not exist in the MotoSimEG Cell file. When G-Code Converter tries to create a model base on a user frame that does not exist, a create Model Error is returned from MotoSimEG | Open the Configuration File and change the User Frame to the correct User Frame number `<ReferenceModel JobTypeMaster="True">UF6</ReferenceModel>` `<ReferenceModel JobTypeSub="True">UF6</ReferenceModel>` |
| Com. error       | ComAutoError = 13 | Points ImporterEG communication error with MotoSimEG possibly due to lack of computer resources - memory, etc. | - Shut down all applications and reopen only the necessary applications  
- Add more Memory to the PC  
- Add more Virtual Memory to the PC  
- Ensure that you have the recommended version of MotoSimEG installed |
### Table 5  G Code Conversion Errors and Possible Solutions

<table>
<thead>
<tr>
<th>General Category</th>
<th>Indication</th>
<th>Error Explanation</th>
<th>Possible Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Setup Error</strong></td>
<td>Model to Small</td>
<td>Configuration file Linear Units are set at Inches and should be Millimeters</td>
<td>Change Configuration file &lt;LengthUnits&gt; to Millimeters &lt;Units&gt; &lt;LengthUnits&gt;Millimeters&lt;/LengthUnits &lt;AngleUnits&gt;Degrees&lt;/AngleUnits &lt;TimeUnits&gt;Minutes&lt;/TimeUnits&gt; &lt;/Units&gt;</td>
</tr>
<tr>
<td><strong>Operational Error</strong></td>
<td>- XMLFileOpenErr=5 - Cannot open the XML File</td>
<td>Another application is using the XML file or G-Code file.</td>
<td>Find the application that is using the XML/GCode file and close it.</td>
</tr>
<tr>
<td><strong>G-Code Errors</strong></td>
<td>- No Points in Robot Jobs - No Conversion Errors</td>
<td>Configuration file is set to look for the Start of G-Code File symbol (typically %)</td>
<td>Add a Start of File Symbol to G-Code File (%) &lt;StartFile&gt;%&lt;/StartFile&gt; OR Remove the &lt;Start ofFile&gt; command from the configuration file</td>
</tr>
<tr>
<td><strong>G-Code Error</strong></td>
<td>- No conversion Errors - Points missing in robot Job</td>
<td>Configuration File is not handling a specific G-Code Command. For example all M10 commands are missing or all G02 positions are missing.</td>
<td>Add G-Code commands to the configuration file For example, if G02 command is not in the configuration file, add it as shown: &lt;GTOInform&gt; &lt;GCODE func=&quot;G02&quot;&gt; &lt;Header&gt;False&lt;/Header&gt; &lt;Description&gt;Circular Motion Clockwise&lt;/Description&gt; &lt;isLiteralInform&gt;True&lt;/isLiteralInform&gt; &lt;MotionFunction&gt;True&lt;/MotionFunction&gt; &lt;/GCODE&gt; &lt;INFORM&gt; &lt;MotionType&gt;Circular&lt;/MotionType&gt; &lt;/INFORM&gt; &lt;/GTOInform&gt;</td>
</tr>
</tbody>
</table>
5.7.1 Finding G Code Errors

5.7.1.1 Determining G-Code Errors From Robot Job Errors

Some G-Code programs can be extremely large. For example, mold programs may contain over 100,000 G-Code instructions. Pinpointing errors in these files can be very difficult. The following procedure provides a method of tracing errors in the output robot JBI files back to the source G-Code line.

<table>
<thead>
<tr>
<th>General Category</th>
<th>Indication</th>
<th>Error Explanation</th>
<th>Possible Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation</td>
<td>HardwareKeyErr=12</td>
<td>Hardware key not installed or Hardware key driver error</td>
<td>- Install Hardware key on the correct port</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Reinstall key drivers - call Motoman Support personal for updated key drivers</td>
</tr>
<tr>
<td>Operational</td>
<td>CloseMSimErr = 11</td>
<td>User closed the MotoSimEG cell file from within MotoSimEG</td>
<td>Close MotoSimEG Cell file from the Points Importer EG application.</td>
</tr>
</tbody>
</table>
| Setup Error      | Circles not being drawn correctly | - CircleFixup turned Off  
- Robot and CNC draw circles differently. Robot Program must add intermediate points on the Circle. | *Open the configuration file and set the <CircleFixUp>True</CircleFixUp> <CircleIJCenter>False</CircleIJCenter> Or <CircleFixUp>True</CircleFixUp> <CircleIJCenter>True</CircleIJCenter> |
|                  |                     | - Circles are to large or small  
- I,J,K vectors can be offsets from the start point or can be in absolute part coordinate system.  
- The CircleIJCenter parameter determines if the circles are absolute or relative. | *Open the configuration file and set the <CircleFixUp>True</CircleFixUp> <CircleIJCenter>False</CircleIJCenter> Or <CircleFixUp>True</CircleFixUp> <CircleIJCenter>True</CircleIJCenter> |
1. With a third party G-Code Editor, add Line numbers to your G-Code file. For example, using the freeware, CNC Syntax Editor, you can use the Menu Tools > Renumber Lines command shown below:

![Image of CNC Syntax Editor](image1)

2. Select the Write Line Numbers in JBI as Comments Option (see Figure below).

![Image of JBI Configuration](image2)

3. Convert Job again.

4. Open the Error Log and find the bad step Number (see example below). In our example, this is step 7.
5. Open the bad Robot JBI and file bad step # and related G Code Line Number (see below). In our example, Step 7 is created by line N48 in the G-Code file.
Double Click to open bad job file.
6. Open the NC File tab in the G-Code Converter (see example below).

1. Type the bad number in the "Find" text box and press [Find].
2. Select the N48 line number in the "Goto in NC" text box and press [Goto In NC File].
3. Examine the line any errors, syntax errors, missing fields, misplaced decimals, etc.

 NOTE: In this example, the X position is 1004.3825 inches and should be 4.3825 inches
4. Edit the G-Code to fix the problem.
5.7.1.2 Dividing the G Code File

In some cases, it may be easier to divide the G-Code file into smaller jobs. For example, you can quickly make a copy of a G-Code job file and then delete ½ of the points. Attempt another conversion and if the conversion fails, repeat this process of elimination until the conversion is successful. The group of points that caused the failure can then be examined to determine the error source.

5.7.1.3 Quick Resolution of Robot Reach Issues

The purpose of the Config. Control Panel is to allow temporary overrides of some G-Code configuration file commands. These temporary overrides can be used to quickly try different cell configurations to see if they resolve robot reach issues. Use this file to try different User Frame locations, Tool TCPs, Initial Robot Pose or to change any of the other parameters listed on this tab.

Note: These changes are temporary changes. Once you determine the optimum conversion parameters, you must modify the G-Code configuration file to make these changes permanent.
5.7.2 Log File

The JBI Log file is another source for possible clues to the causes of errors.
Possible Solutions

- Check that Units are correct (Length and Angle Units)
- Check that User Frame is correct
- Adjust Robot Configuration
  - Change A, B, and C angles
  - Adjust Robot Starting, or 3DOF Angles
  - Adjust Initial robot configuration
- Break NC Job into smaller programs with different starting robot configurations
- Move Robot
- Change Tool or Tool configuration
- Move Part

5.7.3 Job Conversion Reports

The job conversion report can also be used to troubleshoot the G-Code conversion process. The Summary panel provides a place to enter notes and comments regarding the G-Code conversion and create a complete status report. The status report is saved as a .pdf file and contains all information regarding the G-Code conversion including files used and created and any errors that occurred during the process.
Notes
Chapter 6

AutoCAD to Robot Job Converter

The AutoCAD to Robot Job Converter application enables you to import MotoSim EG process paths from AutoCAD. This allows you to create MotoSim EG process paths directly on AutoCAD part models and then import these process paths directly into MotoSim EG.

The figure below provides an overview of the AutoCAD to Robot Job process including intermediate files that are produced.

The AutoCAD file is first converted into a master XML file. This file is then broken into master and callable subroutines XML job files. Each of these files has both the XML header section as well as the point location data. These XML files are now compatible with the main MotoSim EG Points Importer application.

The third and fourth conversions are triggered by the AutoCAD to Robot Job Converter application but are actually performed by the Points Importer and the MotoSim EG applications. These conversions produce an intermediate set of INF files that are used by MotoSim EG to produce the final set of robot JBI job files.

6.1 Installing AutoCAD to Robot Job Converter

The AutoCAD to Robot Job Converter application is a productivity add-on to the MotoSim EG Points Importer client application and therefore requires both the MotoSim EG Points Importer (P/N 148611-2) and MotoSim EG (Full 151021-1 or Lite 151019-1). In addition, AutoCAD to Robot Job Converter requires a full AutoCAD license. All MotoSoft software is provided on a single CD-ROM with a browser/installer utility (Motoman CD-ROM Browser, P/N 141720-1). Please refer to Section 1.5 for detailed installation instructions.
6.2 First Time Use

The AutoCAD to Robot Job Converter application is accessed from MotoSim EG Points Importer using the [AutoCAD to JBI] button on the toolbar.

Before you begin using AutoCAD to Robot Job Converter, you must first have a valid MotoSim EG.mel robot cell file and properly structured AutoCAD files with process paths clearly defined. The MotoSim EG cell must have a robot, TCP or Tool Center Point set, and have a user frame (part) coordinate system setup. After first time use, MotoSim EG Points Importer will automatically start MotoSim EG and open the last cell modified.

6.2.1 Sample Files

AutoCAD to Robot Job Converter includes several sample files that you can open and use to get started. These sample files are located in the Example Cells folder in the MotoSim EG Points Importer folder (c:\Program Files\Motoman\MotoSim EG Points Importer\Example Cells).
6.3  Understanding the AutoCAD to Robot Job Converter Work Area

The AutoCAD to Robot Job Converter application features an easy to use graphical interface. All toolbar button commands can also be found under the File dropdown window. In addition, you can toggle toolbars on and off using the View menu commands.

6.3.1 Command Menus

File
Contains all AutoCAD to Robot Job Converter process commands as listed below.

View
Allows you to toggle the view for each toolbar.

Help
- About - Contains version information.
- Contents - Opens the interactive Help file.
6.3.2 **Configuration Files Toolbar**

The Configuration Files toolbar enables you to determine what Cell, Job, and DWG configuration files will be used to create the robot job file. Each window is color coded to show the status of each file. Red indicates that there is a problem with this file. Yellow indicates that the file has yet to be loaded, and Green means that all files are correct and the application is ready to convert your DWG file. You can also click on the wrench icon to edit any of the configuration files.

⚠️ **WARNING!**
Changes to configuration files do not take effect until files are reloaded.

### 6.3.2.1 XYZ or Relative Job Output Option

To make the output job relative job (Cartesian coordinates), add the following commands to the Job Configuration file. All RobotPosture commands should be set to 0. MotoSimEG will automatically assign them the correct values.

```xml
<JobOutputOptions>
  <JobType>
    <RectanJobType>
      <PositionConfiguration>
        <UserFrame>1</UserFrame>
      </PositionConfiguration>
      <RobotPosture>
        <FlipPosture>0</FlipPosture>
        <UpperPosture>0</UpperPosture>
        <FrontPosture>0</FrontPosture>
        <RPosture>0</RPosture>
        <TPosture>0</TPosture>
      </RobotPosture>
    </RectanJobType>
  </JobType>
</JobOutputOptions>
```

### 6.3.2.2 XYZ or Pulse Job Output Option

Job outputs will produce Pulse type jobs if the following command is added to the Job Configuration file or if the `JobOutputOptions` command is completely removed from the Job Configuration File.

```xml
<JobOutputOptions>
  <JobType><PulseJobType>True</PulseJobType></JobType>
</JobOutputOptions>
```
6.3.2.3 Template Files

Because the AutoCAD paths imported cannot contain extra process information such as IO commands, Arc On/Off, etc., the AutoCAD Converter can also process additional XML Template files. These template files are used to add additional process information to AutoCAD generated files as well as extra locations.

Template files are called or triggered for inclusion from AutoCAD Z Vectors using specific colors. For example, any Z Vector (defined later in the Chapter) that has a Cyan Color will include the ArcOn.XML template file and adds additional commands and/or locations to the this AutoCAD point.

```
<AutoCADPoints>
    <AutoCADColor>Cyan</AutoCADColor>
    <AutoCADName>ArcOn</AutoCADName>
</AutoCADPoints>

<INFORM>
    <TemplateFile>ArcOn.XML</TemplateFile>
</INFORM>
```

Template Files are XML files that have additional $ commands. $ Commands are used as place holders for the real data. For example, in the ArcOn.XML template file shown below, there is a Position tag that will move to the location of the AutoCAD Position in addition to a Z value of 150.00 millimeters. This is because the first Position tag has a $Z $+ $150 field. This is converted to Z + 150.0. The Motion type will be joint motion with a velocity of 80%.

A Call Job TORCHCL and TORCHCK is added to the job with additional position points added after the calls. The next 2 positions are added to the AutoCAD location, along with a Call Job ArcOn and timer delay of 1 second.

```
<Root>
    <Position>
        <MotionType>Joint</MotionType>
        <Velocity>80</Velocity>
        <X>$X</X>
        <Y>$Y</Y>
        <Z>$Z $+ $150</Z>
        <Rx>$RX</Rx>
        <Ry>$RY</Ry>
        <Rz>$RZ</Rz>
    </Position>

    <GenInform>CALL JOB:TORCHCL</GenInform>
    <GenInform>CALL JOB:TORCHCK</GenInform>
    <Timer>1</Timer>

    <Position>
        <MotionType>Joint</MotionType>
        <Velocity>80</Velocity>
        <X>$X</X>
        <Y>$Y</Y>
    </Position>
</Root>
```

Note: This is a very flexible method of adding extra job information that can be used to create search patterns, weave offsets, or Arc On/Off commands. There is no limit to the number of points that can be inserted in this way.

Template Files ArcOn.XML Example File

```
<Root>
    <Position>
        <MotionType>Joint</MotionType>
        <Velocity>80</Velocity>
        <X>$X</X>
        <Y>$Y</Y>
        <Z>$Z $+ $150</Z>
        <Rx>$RX</Rx>
        <Ry>$RY</Ry>
        <Rz>$RZ</Rz>
    </Position>

    <GenInform>CALL JOB:TORCHCL</GenInform>
    <GenInform>CALL JOB:TORCHCK</GenInform>
    <Timer>1</Timer>

    <Position>
        <MotionType>Joint</MotionType>
        <Velocity>80</Velocity>
        <X>$X</X>
        <Y>$Y</Y>
    </Position>
</Root>
```
The AutoCADTemplateDirectory command in the AutoCAD Configuration File defines the directory or location of all template files (See Section 6.3.2.6, "AutoCAD Configuration File").

6.3.2.4 Cell Configuration File

The xml cell configuration file contains data related to the robot and MotoSim EG cel being used for conversion. Allowable fields and values are described in the CelConfigurationSchema(x).xsd (where x is an integer that is incremented each time the schema file is updated). Refer to Appendix F for more information.

Sample Cell Configuration File:

```xml
<Root>
  <Settings>
    <ArmConfig>
      <S>24728</S>
      <L>24728</L>
      <U>24728</U>
      <R>24728</R>
      <B>24728</B>
      <T>24728</T>
    </ArmConfig>
  </Settings>
  <GenInform>CALL JOB:ARCON</GenInform>
  <Timer>1</Timer>
</Root>
```

The Cell Configuration File window allows you to upload and edit the XML cell configuration file you want to use to create your robot job.

To load a cell configuration file:

1. Click on the folder icon.
2. Locate the file you want to use and click Open. If AutoCAD to Robot Job Converter displays the cell file in Red, there may be a problem with the selected file. Make certain you have selected the correct XML cell configuration file, or edit the file to correct the problem.
To edit a cell configuration file:

1. Click on the wrench icon.
2. The Points to Robot Job Converter application opens the XML file in the default XML editor. Remember: Changes to configuration files do not take effect until all file are reloaded.
3. Save your changes to the cell configuration file.
4. Click on the All folder to reload the configuration files and allow changes to take effect.

6.3.2.5 Job Configuration File

The xml job configuration file contains information that describes how the point job will be converted into a robot job. This file defines a number of point related data such as:

- Dimensional units of the data in the point file
- Angle System used to define the robot orientation coordinate system
- Default Motion types and speeds

This file also contains cell data that determines how the points are converted, including:

- User Frame to attach data
- Maximum number of lines in each sub-routine
- Job output options (relative or joint)
- Tool number

Allowable fields and values are described in the JobConfigSchemaV(x).vsd (where x is an integer that is incremented each time the schema file is updated). Refer to Appendix F for more information.

Sample Job Configuration File:

```
<Settings>
  <InformFileFormats MasterandSubs="True" MasterName="gmaster" MaxLines="900"/>
  <AngleSystem>YPR</AngleSystem>
  <Units>
    <LengthUnits>Inches</LengthUnits>
    <AngleUnits>Degrees</AngleUnits>
    <TimeUnits>Minutes</TimeUnits>
  </Units>
  <DefaultVelocity>
    <Linear>50.0</Linear>
    <Joint>5</Joint>
  </DefaultVelocity>
  <DefaultMotionType>Linear</DefaultMotionType>
  <ReferenceModel>UF1</ReferenceModel>
  <Tool>0</Tool>
</Settings>
```

The Job Configuration File window allows you to upload and edit the XML job configuration file you want to use to create your robot job.
To load a job configuration file:

1. Click on the folder icon.
2. Locate the file you want to use and click Open. If AutoCAD to Robot Job Converter displays the cell file in Red, there may be a problem with the selected file. Make certain you have selected the correct XML job configuration file, or edit the file to correct the problem.

To edit a job configuration file:

1. Click on the wrench icon.
2. The AutoCAD to Robot Job Converter application opens the XML file in the default XML editor. Remember: Changes to configuration files do not take effect until all file are reloaded.
3. Save your changes to the job configuration file.
4. Click on the All folder to reload the configuration files and allow changes to take effect.

6.3.2.6 AutoCAD Configuration File

The AutoCAD configuration file contains detailed information regarding the AutoCAD file you are using to create your robot job. Allowable fields and values are described in the DWGConfigurationSchema(x).xsd (where x is an integer that is incremented each time the schema file is updated). Refer to Appendix F for more information.

This file is primarily used to map AutoCAD vectors and colors into Robot Positions and Motion Types. It is also used to include XML Template Files for some Vector colors.

Note: All colors and motion types are referring to the Z Vector defined later in this Chapter. All X Vectors are blue in color by default.

Example AutoCAD Configuration File:

```xml
<Root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:noNamespaceSchemaLocation="C:\PROGRA~1\Motoman\MOTOSI~1\PointsImporterEgschema\AutoCadConverterSchemaV1.xsd">
    <Settings>
        <Units>
            <LengthUnits>Millimeters</LengthUnits>
            <AngleUnits>Degrees</AngleUnits>
            <TimeUnits>Seconds</TimeUnits>
        </Units>
        <AutoCADLayerRootName>PATH</AutoCADLayerRootName>
        <AutoCADTemplates>
            <AutoCADTemplateDirectory>C:\Program Files\Motoman\MotoSim EG\Cells\ACADDemo1\XMLConfigurationFiles\XMLTemplates</AutoCADTemplateDirectory>
        </AutoCADTemplates>
        <ThreeDOFProcessing>
            <UseThreeDOF>False</UseThreeDOF>
        </ThreeDOFProcessing>
    </Settings>
    <AUTOCADTOINFORM>
        <AutoCADPoints>
            <AutoCADColor>Cyan</AutoCADColor>
            <AutoCADName>ArcOn</AutoCADName>
        </AutoCADPoints>
        <INFORM>
            <TemplateFile>ArcOn.XML</TemplateFile>
        </INFORM>
    </AUTOCADTOINFORM>
</Root>
```
The AutoCAD Configuration File window allows you to upload and edit the XML AutoCAD configuration file you want to use to create your robot job.

**To load a AutoCAD configuration file:**

1. Click on the folder icon.

2. Locate the file you want to use and click Open. If AutoCAD to Robot Job Converter displays the cell file in Red, there may be a problem with the selected file. Make certain you have selected the correct XML AutoCAD configuration file, or edit the file to correct the problem.
To edit a AutoCAD configuration file:
1. Click on the wrench icon.
2. The AutoCAD to Robot Job Converter application opens the XML file in the default XML editor. Remember: Changes to configuration files do not take effect until all file are reloaded.
3. Save your changes to the AutoCAD configuration file.
4. Click on the All folder to reload the configuration files and allow changes to take effect.

6.3.2.7 Clear Configuration Files
The Clear Configuration Files button clears all your selections for cell configuration files.

6.3.2.8 Reload Configuration Files
The Reload Configuration Files button reloads all configuration files. It is very important to remember to reload configuration files after making any changes to the configurations files. This updates the configuration files allowing any changes to take effect.

6.3.3 Converter Setup Toolbar
![Converter Setup Toolbar]

The Converter Setup toolbar enables you to select the AutoCAD file you wish to convert to a robot job. Once the file is converted, you can then play the job in MotoSim EG.

6.3.3.1 Select and Convert Points File
![Select and Convert Points File]

The DWG to JBI button allows you to select the DWG file you want to convert to a robot job. To select and convert a points file:
1. Click on the DWG to JBI button.
2. Locate the file you want to convert and click Open. AutoCAD to Robot Job Converter begins processing the files you have selected to create a robot JBI file.
3. Upon success, you will receive a message stating that the AutoCAD file has been successfully converted into a robot job. Click OK.
4. You may now play the converted job in MotoSim EG.

6.3.3.2 Play Job
![Play Job]

The Play JOB button opens the most recently converted job in MotoSim EG and begins playback.

6.3.4 Edit Toolbar
![Edit Toolbar]

Converting a points file to a robot job creates several intermediate process files including XML and INF. The Edit toolbar allows you to open and view each of these intermediate files generated by the AutoCAD to Robot Job Converter. This can be a valuable tool for troubleshooting your job conversion.
You can also view any error files or open the job folder directly to view other files associated with your project.

6.3.4.1 Edit DWG

To view/edit the DWG file:
1. Click on the Edit DWG button.
2. The AutoCAD to Robot Job Converter application opens the DWG file in the AutoCAD application.
3. Save your changes to the DWG configuration file.

6.3.4.2 XML Files

The AutoCAD to Robot Job Converter application creates several XML files during the conversion process. The AutoCAD data is converted into an intermediate XML format creating a Master file and several subroutine file that will ultimately be converted into the .jbi job format.

To view/edit the XML files:
1. Select the XML file from the dropdown menu.
2. Click on the XML Files button.
3. The AutoCAD to Robot Job Converter application opens the XML file in the default XML editor.
4. Save your changes to the XML file.

6.3.4.3 INF Files

The AutoCAD to Robot Job Converter application creates several INF files during the conversion process. The intermediate XML files are converted into the INF format that can be converted by the MotoSim EG application into the .jbi job format.

To view/edit the INF files:
1. Select the INF file from the dropdown menu.
2. Click on the INF Files button.
3. The AutoCAD to Robot Job Converter application opens the INF file in the default text editor.
4. Save your changes to the INF file.

6.3.4.4 JBI Files

To view/edit the JBI files:
1. Select the JBI file from the dropdown menu.
2. Click on the JBI Files button.
3. The AutoCAD to Robot Job Converter application opens the JBI file in the default text editor.
4. Save your changes to the JBI file.
6.3.4.5 View Error File

The View Error File button opens any error file generated by the AutoCAD to Robot Job Converter application during the job conversion process.

6.3.4.6 Job Directory

The JOB DIR. button allows you to browse to and open the job folder directly to view any file associated with your project.

6.3.5 Transparency

The transparency slider allows you to make the MotoSim Points Importer EG work area appear transparent. When this is done, other application windows can be seen through the work area. This can allow you to view a robot job running in the MotoSim EG application window while continuing work in the MotoSim Points Importer EG work area.

6.3.5.1 On Top Check Box

The On Top check box allows you to set the Points to Robot Job Converter application to always appear on top of other tiled windows or applications.

6.3.6 Fault Logging

When Fault Logging is ON, detailed event messages are written to the Windows Event Log call MotomanAutoCADC log as shown below.

*Note: Only events from the last conversion are listed. When a new conversion starts, the old conversion events are erased.*
This information can be saved and emailed to Motoman support personal to help troubleshoot various position data to robot job conversion errors. The screen shot below shows the appropriate menu command used to save the event file.

**To save the log file:**
1. Select Action > Save Log File As... from the dropdown menu.
2. Enter the name you want to save the file as and click OK.

6.4 **AutoCAD to Robot Job Converter Support Files**

The AutoCAD to Robot Job Converter uses several support files to convert the AutoCAD DWG file to a robot JBI file. Besides the XML configuration files, AutoCAD to Robot Converter must have properly structured AutoCAD files with process paths clearly defined. In addition, tool data may need to be modified in the MotoSim EG cel file to reflect Autacoid coordinate conventions.

6.4.1 **Creating Autacoid files with Process Paths**

Process paths are created in Autacoid 3D using path vectors that define tool orientation and motion type. Each point along the process path is made up of two vectors; Z, and X. The Z-vector defines tool posture while the X-vector defines tool orientation. For example, an application such as drilling may require that the tool be perpendicular to the part surface. In this case, the Z-vector would start at a point on the part surface and extend outward to create a 90 degree angle with the part surface. The X-vector would start at the same point on the part surface and extend outward to define the tool orientation.

The Color of the Z-vector determines the motion type (Red for linear motion, Yellow for circular). This element is user controlled and defined in the AutoCADConfigxxxx.xml file.
The example below defines Red as a Linear Motion type.

<AutoCADPoints>
  <AutoCADColor>Red</AutoCADColor>
  <AutoCADName>MoveLinear</AutoCADName>
</AutoCADPoints>

<INFORM>
  <MotionType>Linear</MotionType>
</INFORM>

Circular movements must be defined using at least 3 points. Two arcs that are near each other must be separated by a linear (Red) vector line (see Figure below). The X-vector defines tool orientation and must be blue. This vector corresponds to the robot tool X-coordinate and determines the orientation of the robot tool at a given point. For example, an application such as welding may wish to keep the same tool orientation relative to the part at each process location.

Note: The “X-vector” must be perpendicular to the drawn “Z-vector”. X and Z vectors can be drawn together as a block for placement purposes but must be separated before use in MotoSim EG Points Importer.
Once each process point has been defined with both the X- and Z-vectors, and appropriate Calor for motion type, it is time to connect each point to create the process path. A green 3D polyline is used to connect each process point.

*Note: The 3D polyline Calor must be green. This allows the OLP software to find the path. No other 3D polylines in the drawing can be green.*

The sequence the polyline is drawn is the same sequence the program will be processed when imported into MotoSim EG. Multiple process paths must be created on separate layers and labeled “PATHX” where “X” equals a numeric value (1-?). The name of each layer should correspond to the order in which the path is processed (“Path1”, “Path2,” etc.). This ensures that the paths are processed in the correct sequence.

**To create process paths in AutoCAD:**

1. Begin with an AutoCAD 3D modeled part, drawn in full scale and dimensioned in millimeters.

2. Define the drawing origin or world coordinate system (WCS) at a point on the part that can easily be used for alignment in the production cell. This origin will be placed at the UserFrame origin defined in the MotoSim EG cell. Care must be taken when defining the WCS origin in AutoCAD so that it is placed appropriately within the MotoSim EG cell. Use the following guidelines to define the WCS origin directions:
   - Define Z direction pointing away from the origin and towards the top of the part.
   - Define X direction pointing away from the origin and down the length of the part.

3. Begin defining process points, drawing all vectors with points at the end of each path.

4. Connect each process point with a green 3D polyline in the order the path is to be processed.

5. Draw additional paths on different layers labeled "Path1", "Path2," etc. in the sequence the paths are to be processed.

*Note: Approach and retract positions can be defined in AutoCAD by defining process points off the part and included in the process path with the green 3D polyline. These positions can also be added later in MotoSim EG.*
6.4.2 MotoSim EG Cell Setup

1. Model the robotics cell in MotoSim EG.

2. Generate tool data that corresponds to the production tool data. The convention used in the 3D AutoCAD file defines the Z-vector as extending outward from the part. Given this convention, current tool information must be rotated about the X-axis by 180 deg. The assumption is made that the current roll, pitch, and yaw angles are well defined for the robot tool frame. For example, if the original tool data was:

   X=-40.500, Y=0.000, Z=139.500, RX=0.000, RY=-45,000, RZ=0.000

   the updated tool data would be:

   X=-40.500, Y=0.000, Z=139.500, RX=180.000, RY=-45,000, RZ=0.000

3. Create a User Frame location where the points in the 3D AutoCAD file are relative.
6.5 Getting Started (Quick Start)

This section provides a quick overview describing how to convert a new AutoCAD DWG file to a robot jbi job files using the AutoCAD to Robot Job Converter application. Before using the AutoCAD to Robot Job Converter application, you must first have a valid MotoSim EG .cel robot cell file and user-defined XML configuration files using the correct schema. Several sample files are included with MotoSim Points Importer EG and can be located in c:\Program Files\Motoman\MotoSim Points Importer EG\Cells\AutoCADDemo.

6.5.1 MotoSim EG Cell Setup

Before you can convert any points files, you must have a MotoSim EG cell file created. This cell must include a robot, Tool Center Point (TCP) or TCP set, and a user frame (part) coordinate system setup.
To create a new MotoSim EG Cell:
1. From the MotoSim EG main window, select File > New Cel Project... The New Cel window appears.

![New Cell Window](image)

2. Enter the desired name for the cell in the File Name window and click Open. The new cell is created in MotoSim EG.

To add a robot to the cell file:
1. With the new cell open in MotoSim EG, select Robot > Add Robot... The Open window appears.
2. Browse to the desired robot folder and select the All.prm file and click Open. The Install Robot window appears.
3. If this is the correct robot, click OK. The robot model appears in the cell.

To add TCP offsets:
1. With the cell open and robot model visible, zoom into the T-axis of the robot.
2. Select Robot > Data Setting > Tool Data. The TOOL Editor window appears.

![TOOL Editor](image)

3. Modify the X, Z, and Ry values to reflect your tool.
To add Tool Model (Optional):

An optional tool model can be added to the cell to act as a visual aid for the TCP location.

1. From the MotoSim EG tool bar, select the CAD Tree button. The CAD Tree window appears.
2. With the tree expanded, right click on the robot tcp and select New Model... from the dropdown window. The Add Model Dialog window appears.
3. Enter the name of the tool model and click OK.
4. Click OK again to confirm file creation. The tool model appears in the CAD tree.
5. Double click on the new tool model file in the CAD tree. The tool model edit window appears.

6. From the Add Parts dropdown window, select CONE and click Add. The CONE Edit window appears.
7. Modify the L. Diameter to reduce the diameter of the cone and click OK.
8. Close the Tool Model window.
9. From the CAD Tree, right click on the tool model and select Select Position... from the dropdown window. The Position window appears.

![Position Tool Model](image)

10. Change the Z(mm) value to -100 to place the tip of the cone at the tool frame origin and click OK.
11. Close the CAD Tree view.

**To create a User Frame Model (Optional):**
1. From the MotoSim EG tool bar, select Robot > Reach View > Reach View. The Reach Area window appears.

![Reach Area](image)

2. Select the Tool radio button from the Mode panel.
3. Click Execute.
4. Click OK.
5. With the CAD tree expanded, right click on world and select New Model... from the dropdown window. The Add Model Dialog window appears.
6. Enter the name of the user frame model and click OK.
7. Click OK again to confirm file creation. The user frame model appears in the CAD tree.
8. Double click on the new user frame model file in the CAD tree. The user frame model edit window appears.
9. From the Add Parts dropdown window, select BOX and click Add. The BOX Edit window appears.

10. Modify the width and height to reduce the size of the box and click OK.

11. Close the User Frame Model window.

12. From the CAD Tree, right click on the user frame model and select Select Position... from the dropdown window. The Position window appears.

13. Change the X (mm) and Z(mm) values to place the user frame model in front of the robot model and within the robot reach.

14. Click OK.

15. Close the CAD Tree view.

16. From the MotoSim EG tool bar, select Robot > Reach View > Delete Reach View.

To define the user frame:

1. From the MotoSim EG tool bar, select Robot > Data Setting > Set User Frame... The Set User Frame window appears.

2. Using the cursor, select the corner of the user frame model.

3. De-select the Active radio button and click Close.

To save the cell file:

1. From the MotoSim EG tool bar, select File > Save.
6.5.2 AutoCAD File Conversion

To setup initial arm configuration in XML configuration file:

1. From the MotoSim EG main window, click on the Position and Job View button from the Main tool bar. The Position and Job windows appear.
2. From the Position window, click on the Frame button.

3. Select the Pulse radio button from the Display Reference Frame window and click OK.

4. Locate the ArmConfig element in the XML Configuration file.

   <ArmConfig>
   <S>-6733</S>
   <L>118307</L>
   <U>87216</U>
   <R>-20336</R>
   <B>13994</B>
   <T>10689</T>
   </ArmConfig>

5. Copy and Paste the pulse position data for each of the axes (SLURBT) into the S, L, U, R, B and T child elements.
6.5.3 Job Configuration Setup

To set units in Job configurations file:
The measurement units used in the XML Configuration file must match those in the points file.

1. Open the XML Configuration file in an XML editor or text editor such as Notepad.
2. Locate the Units element.
   ```xml
   <Units>
       <LengthUnits>Millimeters</LengthUnits>
       <AngleUnits>Degrees</AngleUnits>
       <TimeUnits>Minutes</TimeUnits>
   </Units>
   ```
3. Change the LengthUnits element to either Millimeters or Inches.
4. Change the AngleUnits element to either Degrees or Radians.
5. Change the TimeUnits element to either Minutes or Seconds.

To setup user frame in XML configuration file:

1. Locate the ReferenceModel element and record the User Frame (UF1-UF10).
   ```xml
   <ReferenceModel>UF1</ReferenceModel>
   ```

To make the output job a relative job (Cartesian coordinates), add the following commands to the configuration file:

All RobotPosture commands should be set to 0. MotoSimEG will automatically assign them the correct values.

```xml
<JobOutputOptions>
    <JobType>
        <RectanJobType>
            <PositionConfiguration>
                <UserFrame>1</UserFrame>
            </PositionConfiguration>
            <RobotPosture>
                <FlipPosture>0</FlipPosture>
                <UpperPosture>0</UpperPosture>
                <FrontPosture>0</FrontPosture>
                <RPosture>0</RPosture>
                <TPosture>0</TPosture>
            </RobotPosture>
        </RectanJobType>
    </JobType>
</JobOutputOptions>
```

Note: All RobotPosture commands should be set to 0. MotoSimEG will automatically assign them the correct values.
To make the output job a joint job:

1. Locate the JobOutputOptions command and add a PulseJobType Command with a True datafield as shown below.
2. Or delete the JobOutputOptions command and sub commands for the configuration file.

   <JobOutputOptions>
   <JobType><PulseJobType>True</PulseJobType></JobType>
   </JobOutputOptions>

6.5.4 AutoCAD Configuration Setup

To define the base AutoCADLayer Name:

Locate the AutoCADLayerRootName element and set it the same label that is used in AutoCAD to define the part path layers.

For example, if an AutoCAD dwg had 10 paths that you wanted to transform into a robot job, and are labeled PATH1 to PATH10, then you would insert "PATH" in the AutoCADLayerRootName element.

   AutoCADLayerRootName>PATH</AutoCADLayerRootName>

To map AutoCAD Colors to Motion Types and Templates:

1. Locate the desired AutoCADPoints element or create a new one:
2. In the AutoCADColor place the desired AutoCAD color (legal colors are defined in the associated Schema File.
3. In the AutoCADName element add a descriptive name for the robot motion type. This element is for the operator and is not restricted to a predefined types.
4. In the MotionType element add the associated robot motion type. Legal values are listed in the associated configuration file.
5. Option -an XML template file can also be associated with this color of vector. To define this association, simply add a TemplateFile tag and fill in the template file name.

   <AUTOCADTOINFORM>
   <AutoCADPoints>
      <AutoCADColor>Yellow</AutoCADColor>
      <AutoCADName>MoveCircular</AutoCADName>
   </AutoCADPoints>
   <INFORM>
      <MotionType>Circular</MotionType>
      <TemplateFile>ArcOn.XML</TemplateFile>
   </INFORM>
   <AutoCADPoints>

Note: Multiple AutoCADPoints commands can be used in a single configuration file.
To create a XYZ only AUTOCAD conversion, add a ThreeDOF command as shown below:

1. Locate the or add a ThreeDOF command as shown below:
2. Add the desired values to the Rx, Ry and Rz commands.

```xml
<ThreeDOF>
  <Rx>0</Rx>
  <Ry>0</Ry>
  <Rz>0</Rz>
</ThreeDOF>
```

*Note: Generally the robot tool and TCP are setup so that Rx, Ry and Rz of 0,0,0 will set make the last length of robot tool will be perpendicular to the User frame.*

### 6.5.5 AutoCAD File Conversion

1. If the AutoCAD to Robot Job Converter application is not already running, open it from the MotoSim Points Importer EG command menu.
2. Begin by loading your XML configuration files.
   a. Click on the folder icon for the Cell configuration file.
   b. Locate the file you want to use and click Open. If AutoCAD to Robot Job Converter displays the cell file in Red, there may be a problem with the selected file. Make certain you have selected the correct XML cell configuration file, or edit the file to correct the problem.
   c. Click on the folder icon for the Job configuration file.
   d. Locate the file you want to use and click Open. If AutoCAD to Robot Job Converter displays the job file in Red, there may be a problem with the selected file. Make certain you have selected the correct XML cell configuration file, or edit the file to correct the problem.
   e. Click on the folder icon for the DWG configuration file.
   f. Locate the file you want to use and click Open. If AutoCAD to Robot Job Converter displays the cell file in Red, there may be a problem with the selected file. Make certain you have selected the correct XML cell configuration file, or edit the file to correct the problem.
3. Click on the DWG to JBI button from the Converter Setup toolbar.
4. Locate the DWG file you want to convert and click Open. AutoCAD to Robot Job Converter begins processing the files you have selected to create a robot JBI file.
5. Upon success, you will receive a message stating that the DWG file has been successfully converted into a robot job. Click OK.
6. You may now play the converted job in MotoSim EG.
Notes
Appendix A

Basic Introduction to G-Code

G-code is a common name for the programming language that drives NC and CNC machine tools. While industry has tried to standardize on this basic set of codes, extensions and variations have been added to it independently by manufacturers reflecting the variety of machine tool configurations. Operators must therefore learn the dialects and quirks of the particular machines they use.

G-code is so named because the codes used in a CNC program begin with the letter G. These codes tell the machine tool what type of action to perform, such as:

- Rapid move
- Controlled feed move in a straight line or arc
- Series of controlled feed moves that would result in a hole being bored, a workpiece cut (routed) to a specific dimension, or a decorative profile shape added to the edge of a workpiece.
- Change a pallet
- Set tool information such as offset.

There are other codes including type codes that can be thought of like registers in a computer and provide quick access to commonly used values. These include:

- X position
- Y position
- Z position
- A position (rotary around X)
- B position (rotary around Y)
- C position (rotary around Z)

M Codes (Miscellaneous functions) control the overall machine, causing it to stop, start, turn on coolant, etc. Other codes control the path of the cutting tools. M-Codes include:

- F feed rate
- S spindle speed
- N line number
- R Arc radius
- T Tool selection
- I Arc data X axis
- J Arc data Y axis.
- K Arc data Z axis.
- D Cutter diameter/radius offset
- H Tool length offset

**CAUTION!**
Different machine tools may use the same code to perform different functions: even machines that use the same CNC control.

**Table 1** Common G-Codes

<table>
<thead>
<tr>
<th>G</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G00</td>
<td>Fast positioning</td>
</tr>
<tr>
<td>G01</td>
<td>Linear interpolation</td>
</tr>
<tr>
<td>G02</td>
<td>CW circular interpolation</td>
</tr>
<tr>
<td>G03</td>
<td>CCW circular interpolation</td>
</tr>
<tr>
<td>G10/G11</td>
<td>Data writing/Data write cancel</td>
</tr>
<tr>
<td>G17</td>
<td>X-Y plane selection</td>
</tr>
<tr>
<td>G18</td>
<td>X-Z plane selection</td>
</tr>
<tr>
<td>G19</td>
<td>Y-Z plane selection</td>
</tr>
<tr>
<td>G20</td>
<td>Programming in inches</td>
</tr>
<tr>
<td>G21</td>
<td>Programming in mm</td>
</tr>
<tr>
<td>G28</td>
<td>Return to home position</td>
</tr>
<tr>
<td>G31</td>
<td>Skip function (used for probes and tool length measurement systems)</td>
</tr>
<tr>
<td>G33</td>
<td>Constant pitch threading</td>
</tr>
<tr>
<td>G34</td>
<td>Variable pitch threading</td>
</tr>
<tr>
<td>G40</td>
<td>Tool radius compensation off</td>
</tr>
<tr>
<td>G41</td>
<td>Tool radius compensation left</td>
</tr>
<tr>
<td>G42</td>
<td>Tool radius compensation right</td>
</tr>
</tbody>
</table>
### Table 1  Common G-Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G90</td>
<td>Absolute programming</td>
</tr>
<tr>
<td>G91</td>
<td>Incremental programming</td>
</tr>
<tr>
<td>G94/G95</td>
<td>Inch per minute/Inch per revolution feed</td>
</tr>
<tr>
<td>G96/G97</td>
<td>Constant cutting speed (Constant surface speed)/Constant rotation speed (constant RPM)</td>
</tr>
</tbody>
</table>

### Table 2  Common M-Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M02</td>
<td>Program Stop</td>
</tr>
<tr>
<td>M03</td>
<td>Spindle CW</td>
</tr>
<tr>
<td>M04</td>
<td>Spindle CCW</td>
</tr>
<tr>
<td>M05</td>
<td>Spindle OFF</td>
</tr>
<tr>
<td>M08/M09</td>
<td>Coolant/lubricant ON, Coolant/lubricant OFF</td>
</tr>
<tr>
<td>M30</td>
<td>End of Tape - End of Job</td>
</tr>
</tbody>
</table>
Notes
Appendix B

MotoCal Calibration Tasks For Offline Filter Creation

Jobs created in MotoSim EGEG are based on an ideal Robot.

MotoSim EGEG Simulation Tasks Part 1

1. Create a MotoSim EGEG Cell File.
2. Add a Robot to the Simulation cell.
3. Create a CAD Model of the Workplaces and Holes and Circles (optional).
4. Place the Work Plate Model in the Simulation Cell.
5. Create an ideal user frame with the origin at the first hole.
6. Add an approximate TCP to the Robot.

B.0.1 G-Code Editor Part Programming Tasks

1. Use hole location data (From Coordinate Measuring Machine) to create G-Code jobs that would Move the robot to each of the 4 calibration holes on the work plate.
2. Use G-Code converter to transform the jobs into robot jobs or *.jbi’s.

B.0.2 Robot, Tool and Work piece Calibration

Calibrate the robot, Tool and Work piece (User Frame or Work Calibration) with a Dynalog Encoder and MotoCal Software.

Major Steps
1. Create Robot jobs
2. Attach Universal TCP adaptor to the tool
3. Place Encoder at 1 of 4 locations on the Work Plate.
   a. Attach Encoder to TCP Adaptor.
   b. Move the robot to each taught location and take encoder measurements.
4. Transfer robot jobs to PC.
5. Create a MotoCal Project and created the robot filter
B.0.3 MotoSim EGEG Simulation Tasks - Part 2

1. Select the 4 Hole job as the current program in MotoSim EGEG.
2. Move to each hole and record the location of each hole in Robot Coordinates and enter those values in the following MotoCal Form:

![MotoCal Form](image)

Note: Only use the X, Y, and Z coordinates. R1, R2, and R3 are the rotations of the Work Calibration plate in robot coordinates.

B.0.4 Filter Jobs

1. Use MotoCal to Filter Jobs - Refer to the MotoCal User’s manual (P/N 140724-1) for detailed instructions on how to create and use the filter.
Notes
Appendix C

XML Schema Reference

This chapter explains the governing XML schema files. XML Schema files are files that define basic structure (allowable fields and values) of the intermediate XML Job files.
Schema **AutoCadConverterSchemaV1.xsd**

**Schema location:**

**Attribute form default:**

**Element form default:**

**Elements**

- **AngleUnits**
- **AutoCADColor**
- **AutoCADLayerRootName**
- **AutoCADName**
- **AutoCADPoints**
- **AutoCADTemplateDirectory**
- **AutoCADTempates**
- **AUTOCADTOINFORM**
  - **INFORM**
- **LengthUnits**
- **MotionType**
- **Root**
- **Settings**
- **TemplateFile**
- **ThreeDOFProcessing**
- **TimeUnits**
- **Units**
- **UseThreeDOF**

**element AngleUnits**

<table>
<thead>
<tr>
<th>Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="AngleUnits Diagram" /></td>
</tr>
</tbody>
</table>

**Type:** restriction of xs:string

**Properties:** content simple

**Used by:** element Units

**Facets:**
- enumeration Degrees
- enumeration Radians

**Source:**

```xml
<xs:element name="AngleUnits">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Degrees"/>
      <xs:enumeration value="Radians"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**element AutoCADColor**
<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="AutoCADColor" /></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>type</strong></td>
<td>restriction of xs:string</td>
</tr>
<tr>
<td><strong>properties</strong></td>
<td>content simple</td>
</tr>
<tr>
<td><strong>used by</strong></td>
<td>element AutoCADPoints</td>
</tr>
<tr>
<td><strong>facets</strong></td>
<td>enumeration Cyan, enumeration Red, enumeration White, enumeration Yellow, enumeration Green, enumeration Blue, enumeration Magenta, enumeration 1, enumeration 2, enumeration 3, enumeration 4, enumeration 5, enumeration 6, enumeration 7, enumeration 8, enumeration 9, enumeration 10, enumeration 11, enumeration 12, enumeration 13, enumeration 14, enumeration 15, enumeration 16, enumeration 17, enumeration 18, enumeration 19, enumeration 20, enumeration 21, enumeration 22, enumeration 23, enumeration 24, enumeration 25, enumeration 26, enumeration 27, enumeration 28, enumeration 29, enumeration 30</td>
</tr>
</tbody>
</table>
<xs:element name="AutoCADColor">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Cyan"/>
      <xs:enumeration value="Red"/>
      <xs:enumeration value="White"/>
      <xs:enumeration value="Yellow"/>
      <xs:enumeration value="Green"/>
      <xs:enumeration value="Blue"/>
      <xs:enumeration value="Magenta"/>
      <xs:enumeration value="1"/>
      <xs:enumeration value="2"/>
      <xs:enumeration value="3"/>
      <xs:enumeration value="4"/>
      <xs:enumeration value="5"/>
      <xs:enumeration value="6"/>
      <xs:enumeration value="7"/>
      <xs:enumeration value="8"/>
      <xs:enumeration value="9"/>
      <xs:enumeration value="10"/>
      <xs:enumeration value="11"/>
      <xs:enumeration value="12"/>
      <xs:enumeration value="13"/>
      <xs:enumeration value="14"/>
      <xs:enumeration value="15"/>
      <xs:enumeration value="16"/>
      <xs:enumeration value="17"/>
      <xs:enumeration value="18"/>
      <xs:enumeration value="19"/>
      <xs:enumeration value="20"/>
      <xs:enumeration value="21"/>
      <xs:enumeration value="22"/>
      <xs:enumeration value="23"/>
      <xs:enumeration value="24"/>
      <xs:enumeration value="25"/>
      <xs:enumeration value="16"/>
      <xs:enumeration value="27"/>
      <xs:enumeration value="28"/>
      <xs:enumeration value="29"/>
      <xs:enumeration value="30"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
### element AutoCADLayerRootName

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="AutoCADLayerRootName" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:string</td>
</tr>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>element Settings</td>
</tr>
</tbody>
</table>
| facets | pattern \w{1,10}  
enumeration PATH |
| source  | `<xs:element name="AutoCADLayerRootName">`  
`<xs:simpleType>`  
`<xs:restriction base="xs:string">`  
`<xs:pattern value="\w{1,10}"/>`  
`<xs:enumeration value="PATH"/>`  
`</xs:restriction>`  
`</xs:simpleType>`  
`</xs:element>` |

### element AutoCADName

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="AutoCADName" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:string</td>
</tr>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>element AutoCADPoints</td>
</tr>
</tbody>
</table>
| facets | enumeration ArcOff  
enumeration ArcOn  
enumeration MoveCircular  
enumeration MoveJoint  
enumeration MoveLinear |
| source  | `<xs:element name="AutoCADName">`  
`<xs:simpleType>`  
`<xs:restriction base="xs:string">`  
`<xs:enumeration value="ArcOff"/>`  
`<xs:enumeration value="ArcOn"/>`  
`<xs:enumeration value="MoveCircular"/>`  
`<xs:enumeration value="MoveJoint"/>`  
`<xs:enumeration value="MoveLinear"/>`  
`</xs:restriction>`  
`</xs:simpleType>`  
`</xs:element>` |

### element AutoCADPoints
**element** AutoCADTempateDirectory

<table>
<thead>
<tr>
<th>diagram</th>
<th><a href="#">Diagram</a></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>element <strong>AutoCADTempates</strong></td>
</tr>
<tr>
<td>source</td>
<td><code>&lt;xs:element name=&quot;AutoCADTempateDirectory&quot; type=&quot;xs:string&quot;/&gt;</code></td>
</tr>
</tbody>
</table>

**element** AutoCADTempates

<table>
<thead>
<tr>
<th>diagram</th>
<th><a href="#">Diagram</a></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>properties</th>
<th>content complex</th>
</tr>
</thead>
<tbody>
<tr>
<td>children</td>
<td><strong>AutoCADTempateDirectory</strong></td>
</tr>
<tr>
<td>used by</td>
<td>element <strong>Settings</strong></td>
</tr>
</tbody>
</table>
element **AUTOCADTOINFORM**

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="AUTOCADTOINFORM Diagram" /></td>
</tr>
</tbody>
</table>

properties content complex

children **AutoCADPoints INFORM**

used by element **Root**

source

```xml
dom:element name="AUTOCADTOINFORM">  
  dom:complexType>  
    dom:all>  
      dom:element ref="AutoCADPoints"/>  
      dom:element ref="INFORM"/>  
    </dom:all>  
  </dom:complexType>  
</dom:element>
```

---

element **INFORM**

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="INFORM Diagram" /></td>
</tr>
</tbody>
</table>

properties content complex

children **MotionType TemplateFile**

used by element **AUTOCADTOINFORM**

source

```xml
dom:element name="INFORM">  
  dom:complexType>  
    dom:all>  
      dom:element ref="MotionType"/>  
      dom:element ref="TemplateFile"/>  
    </dom:all>  
  </dom:complexType>  
</dom:element>
```
element **LengthUnits**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="LengthUnits" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of (xs:string)</td>
</tr>
<tr>
<td>properties</td>
<td>content (simple)</td>
</tr>
<tr>
<td>used by</td>
<td>element <strong>Units</strong></td>
</tr>
<tr>
<td>facets</td>
<td>enumeration Millimeters</td>
</tr>
<tr>
<td></td>
<td>enumeration Inches</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;LengthUnits&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:simpleType</a></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:restriction base=&quot;xs:string&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;Millimeters&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;Inches&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:restriction&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:simpleType&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:element&gt;</td>
</tr>
</tbody>
</table>

element **MotionType**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="MotionType" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of (xs:string)</td>
</tr>
<tr>
<td>properties</td>
<td>content (simple)</td>
</tr>
<tr>
<td>used by</td>
<td>element <strong>INFORM</strong></td>
</tr>
<tr>
<td>facets</td>
<td>enumeration Circular</td>
</tr>
<tr>
<td></td>
<td>enumeration Joint</td>
</tr>
<tr>
<td></td>
<td>enumeration Linear</td>
</tr>
</tbody>
</table>
<xs:element name="MotionType">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Circular"/>
      <xs:enumeration value="Joint"/>
      <xs:enumeration value="Linear"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

element Root

diagram

properties content complex

children Settings AUTOCADTOINFORM

source
<xs:element name="Root">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Settings"/>
      <xs:element ref="AUTOCADTOINFORM" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

element Settings

diagram

properties content complex

children Units AutoCADLayerRootName AutoCADTempates ThreeDOFProcessing

used by element Root
<xs:element name="Settings">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Units"/>
      <xs:element ref="AutoCADLayerRootName"/>
      <xs:element ref="AutoCADTemplates"/>
      <xs:element ref="ThreeDOFProcessing"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

element TemplateFile

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>element INFORM</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;TemplateFile&quot; type=&quot;xs:string&quot;/&gt;</td>
</tr>
</tbody>
</table>

element ThreeDOFProcessing

diagram

| properties | content complex |
| children | UseThreeDOF |
| used by | element Settings |
| source  | <xs:element name="ThreeDOFProcessing">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="UseThreeDOF" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

element TimeUnits

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content: simple</td>
</tr>
<tr>
<td>------------</td>
<td>----------------</td>
</tr>
<tr>
<td>used by</td>
<td>element: Units</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration: Seconds, Minutes</td>
</tr>
</tbody>
</table>
| source     | `<xs:element name="TimeUnits">`<xs:simpleType>
  `<xs:restriction base="xs:string">`
  `<xs:enumeration value="Seconds"/>`
  `<xs:enumeration value="Minutes"/>`
  `<xs:restriction>`
  `<xs:simpleType>`
  `<xs:element>` |

**element Units**

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Diagram Units]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>properties</th>
<th>content: complex</th>
</tr>
</thead>
<tbody>
<tr>
<td>children</td>
<td>LengthUnits, AngleUnits, TimeUnits</td>
</tr>
<tr>
<td>used by</td>
<td>element: Settings</td>
</tr>
</tbody>
</table>
| source     | `<xs:element name="Units">`<xs:complexType>
  `<xs:sequence>`
  `<xs:element ref="LengthUnits"/>`
  `<xs:element ref="AngleUnits"/>`
  `<xs:element ref="TimeUnits"/>`
  `<xs:sequence>`
  `<xs:complexType>`
  `<xs:element>` |

**element UseThreeDOF**

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Diagram UseThreeDOF]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content: simple</td>
</tr>
<tr>
<td>used by</td>
<td>element <strong>ThreeDOFProcessing</strong></td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration False</td>
</tr>
<tr>
<td></td>
<td>enumeration True</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>source</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>&lt;xs:element name=&quot;UseThreeDOF&quot;&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:simpleType&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:restriction base=&quot;xs:string&quot;&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;False&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;True&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/xs:restriction&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/xs:simpleType&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/xs:element&gt;</code></td>
</tr>
</tbody>
</table>
element **AngleSystem**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="AngleSystem" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of <em>xs:string</em></td>
</tr>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>element <strong>Root/Settings</strong></td>
</tr>
<tr>
<td>facets</td>
<td>enumeration <strong>YPR</strong></td>
</tr>
<tr>
<td></td>
<td>enumeration <strong>Euler</strong></td>
</tr>
<tr>
<td>annotation</td>
<td>documentation</td>
</tr>
<tr>
<td></td>
<td>This element specifies the reference system for position orientation angles. Values are YPR and Euler. YPR is the standard MotoSimEG angle system. Euler means that the angle system is specified in terms of the Euler Angle system. This is a system that is used in the aircraft industry.</td>
</tr>
</tbody>
</table>
### AngleSystem

```
<xs:element name="AngleSystem">
  <xs:annotation>
    <xs:documentation>
      This element specifies the reference system for position orientation angles. Values are YPR and Euler. YPR is the standard MotoSimEG angle system. Euler means that the angle system is specified in terms of the Euler Angle system. This is a system that is used in the aircraft industry.
    </xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="YPR"/>
      <xs:enumeration value="Euler"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

### AngleUnits

<table>
<thead>
<tr>
<th>property</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:string</td>
</tr>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>element Units</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration Degrees</td>
</tr>
<tr>
<td></td>
<td>enumeration Radians</td>
</tr>
</tbody>
</table>

```
<xs:element name="AngleUnits">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Degrees"/>
      <xs:enumeration value="Radians"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

### DefaultMotionType

```
<xs:element name="DefaultMotionType">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:documentation>
        When a position command does not include a MotionType command, the DefaultMotionType is used.
      </xs:documentation>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```
When a position command does not include a MotionType command, the DefaultMotionType is used.

```xml
<xs:element name="DefaultMotionType">
  <xs:annotation>
    <xs:documentation>When a position command does not include a MotionType command, the DefaultMotionType is used.</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Linear"/>
      <xs:enumeration value="Circular"/>
      <xs:enumeration value="ExternalRef"/>
      <xs:enumeration value="Spline"/>
      <xs:enumeration value="Inc"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**element DefaultVelocity**

- **content**: complex
- **children**: Linear Joint ExternalRef Circular Spline

Default Velocity is the velocity that is used when a position command does not include a velocity command.
**source**

```xml
<xs:element name="DefaultVelocity">
  <xs:annotation>
    <xs:documentation>Default Velocity is the velocity that is used when a position command does not include a velocity command.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Linear" minOccurs="0"/>
      <xs:element ref="Joint" minOccurs="0"/>
      <xs:element name="ExternalRef" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:decimal">
            <xs:fractionDigits value="10"/>
            <xs:totalDigits value="16"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Circular" type="xs:anySimpleType" minOccurs="0"/>
      <xs:element name="Spline" type="xs:anySimpleType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**element DefaultVelocity/ExternalRef**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="Image" alt="Diagram of DefaultVelocity/ExternalRef" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of <code>xs:decimal</code></td>
</tr>
</tbody>
</table>
| properties | isRef 0  
    minOcc 0  
    maxOcc 1  
    content simple |
| facets | totalDigits 16  
    fractionDigits 10 |

**source**

```xml
<xs:element name="ExternalRef" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:decimal">
      <xs:fractionDigits value="10"/>
      <xs:totalDigits value="16"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**element DefaultVelocity/Circular**
### element `Circular`

- **Diagram:** ![Circular Diagram](image)
- **Type:** `xs:anySimpleType`
- **Properties:**
  - `isRef`: 0
  - `minOcc`: 0
  - `maxOcc`: 1
  - `content`: simple
- **Source:**

```xml
<xos:element name="Circular" type="xs:anySimpleType" minOccurs="0"/>
```

### element `Spline`

- **Diagram:** ![Spline Diagram](image)
- **Type:** `xs:anySimpleType`
- **Properties:**
  - `isRef`: 0
  - `minOcc`: 0
  - `maxOcc`: 1
  - `content`: simple
- **Source:**

```xml
<xos:element name="Spline" type="xs:anySimpleType" minOccurs="0"/>
```

### element `Joint`

- **Diagram:** ![Joint Diagram](image)
- **Type:** restriction of `xs:integer`
- **Properties:**
  - `content`: simple
  - `default`: 20
- **Used by:** element `DefaultVelocity`
- **Facets:**
  - `totalDigits`: 3
  - `fractionDigits`: 0
  - `pattern`: \d{1,3}
- **Source:**

```xml
<xos:element name="Joint" default="20">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:fractionDigits value="0"/>
      <xs:totalDigits value="3"/>
      <xs:pattern value="\d{1,3}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```
### element **LengthUnits**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="LengthUnits" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:string</td>
</tr>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>element <strong>Units</strong></td>
</tr>
</tbody>
</table>
| facets   | enumeration Inches  
enumeration Millimeters |
| source   | `<xs:element name="LengthUnits">  
  <xs:simpleType>  
  <xs:restriction base="xs:string">  
    <xs:enumeration value="Inches"/>  
    <xs:enumeration value="Millimeters"/>  
  </xs:restriction>  
</xs:simpleType>` |

### element **Linear**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="Linear" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:decimal</td>
</tr>
</tbody>
</table>
| properties | content simple  
default 100.0 |
| used by  | element **DefaultVelocity** |
| facets   | totalDigits 16  
fractionDigits 10  
pattern \((\d{1,6})\).*\d{0,10} |
| source   | `<xs:element name="Linear" default="100.0">  
  <xs:simpleType>  
  <xs:restriction base="xs:decimal">  
    <xs:totalDigits value="16"/>  
    <xs:fractionDigits value="10"/>  
    <xs:pattern value="(\d{1,6})\.*\d{0,10}"/>  
  </xs:restriction>  
</xs:simpleType>` |

### element **ReferenceModel**
element **ReferenceModel**

```xml
<xs:element name="ReferenceModel" default="UF1">
  <xs:annotation>
    <xs:documentation>Specifies the user frame number or MotoSimEG Cell Model that is used to specify the origin of the part coordinate system. </xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value=".{1,15}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**element** **RobotPosture**

```xml
<xs:element name="RobotPosture">
  <xs:complexType>
    <xs:children>
      <xs:element name="FlipPosture"/>
      <xs:element name="UpperPosture"/>
      <xs:element name="FrontPosture"/>
      <xs:element name="RPosture"/>
      <xs:element name="TPosture"/>
    </xs:children>
  </xs:complexType>
</xs:element>
```
<xs:element name="RobotPosture">
  <xs:complexType>
    <xs:all>
      <xs:element name="FlipPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="UpperPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="FrontPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="RPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="TPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:all>
  </xs:complexType>
</xs:element>
### element RobotPosture/FlipPosture

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="FlipPosture" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:integer</td>
</tr>
</tbody>
</table>
| properties | isRef 0  
|          | content simple  
|          | default 0         |
| facets  | enumeration 0  
|          | enumeration 1     |
| source  | `<xs:element name="FlipPosture" default="0">  
|          |   `<xs:simpleType>  
|          |     `<xs:restriction base="xs:integer">  
|          |       `<xs:enumeration value="0"/>  
|          |       `<xs:enumeration value="1"/>  
|          |     </xs:restriction>  
|          |   </xs:simpleType>  
|          | </xs:element>` |

### element RobotPosture/UpperPosture

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="UpperPosture" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:integer</td>
</tr>
</tbody>
</table>
| properties | isRef 0  
|          | content simple  
|          | default 0         |
| facets  | enumeration 0  
|          | enumeration 1     |
| source  | `<xs:element name="UpperPosture" default="0">  
|          |   `<xs:simpleType>  
|          |     `<xs:restriction base="xs:integer">  
|          |       `<xs:enumeration value="0"/>  
|          |       `<xs:enumeration value="1"/>  
|          |     </xs:restriction>  
|          |   </xs:simpleType>  
|          | </xs:element>` |

### element RobotPosture/FrontPosture

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="FrontPosture" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:integer</td>
</tr>
</tbody>
</table>
element RobotPosture/RPosture

diagram

```
<xs:simpleType>
  <xs:restriction base="xs:integer">
    <xs:enumeration value="0"/>
    <xs:enumeration value="1"/>
  </xs:restriction>
</xs:simpleType>
```

type restriction of xs:integer

properties

<table>
<thead>
<tr>
<th>isRef</th>
<th>content</th>
<th>default</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>simple</td>
<td>0</td>
</tr>
</tbody>
</table>

facets

| enumeration 0 | enumeration 1 |

source

```
<xs:element name="RPosture" default="0">
  <xs:restriction base="xs:integer">
    <xs:enumeration value="0"/>
    <xs:enumeration value="1"/>
  </xs:restriction>
</xs:element>
```

element RobotPosture/TPosture

diagram

```
<TPosture/>
```

type restriction of xs:integer

properties

<table>
<thead>
<tr>
<th>isRef</th>
<th>content</th>
<th>default</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>simple</td>
<td>0</td>
</tr>
</tbody>
</table>

facets

| enumeration 0 | enumeration 1 |

source

```
<xs:element name="TPosture" default="0">
  <xs:restriction base="xs:integer">
    <xs:enumeration value="0"/>
    <xs:enumeration value="1"/>
  </xs:restriction>
</xs:element>
```
element Root

**diagram**

![Diagram of Root and Settings elements]

**properties** content complex

**children** Settings

**annotation** documentation

Comment describing your root element

**source**

```xml
<xs:element name="Root">
  <xs:annotation>
    <xs:documentation>Comment describing your root element</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Settings">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="InformFileFormats" type="InformFileFormatsType"/>
            <xs:element ref="AngleSystem"/>
            <xs:element ref="Units"/>
            <xs:element ref="DefaultVelocity"/>
            <xs:element ref="ThreeDOF" minOccurs="0"/>
            <xs:element ref="DefaultMotionType"/>
            <xs:element ref="ReferenceModel"/>
            <xs:element name="ToolNumber" default="0">
              <xs:simpleType>
                <xs:restriction base="xs:int">
                  <xs:enumeration value="0"/>
                  <xs:enumeration value="1"/>
                  <xs:enumeration value="2"/>
                  <xs:enumeration value="3"/>
                  <xs:enumeration value="4"/>
                  <xs:enumeration value="5"/>
                  <xs:enumeration value="6"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```
element Root/Settings

diagram

InformFileFormats

AngleSystem

This element specifies the reference system for position orientation angles. Values are YPR and Euler. YPR is the standard MotoSimEG angle system. Euler means that the angle system is specified in terms of the Euler Angle system. This is a system that is used in the aircraft industry.

Units
<xs:element name="Settings">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="InformFileFormats" type="InformFileFormatsType"/>
      <xs:element ref="AngleSystem"/>
      <xs:element ref="Units"/>
      <xs:element ref="DefaultVelocity"/>
      <xs:element ref="ThreeDOF" minOccurs="0"/>
      <xs:element ref="DefaultMotionType"/>
      <xs:element ref="ReferenceModel"/>
      <xs:element name="ToolNumber" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:int">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
            <xs:enumeration value="2"/>
            <xs:enumeration value="3"/>
            <xs:enumeration value="4"/>
            <xs:enumeration value="5"/>
            <xs:enumeration value="6"/>
            <xs:enumeration value="7"/>
            <xs:enumeration value="8"/>
            <xs:enumeration value="9"/>
            <xs:enumeration value="10"/>
            <xs:enumeration value="11"/>
            <xs:enumeration value="12"/>
            <xs:enumeration value="13"/>
            <xs:enumeration value="14"/>
            <xs:enumeration value="15"/>
            <xs:enumeration value="16"/>
            <xs:enumeration value="17"/>
            <xs:enumeration value="18"/>
            <xs:enumeration value="19"/>
            <xs:enumeration value="20"/>
            <xs:enumeration value="21"/>
            <xs:enumeration value="22"/>
            <xs:enumeration value="23"/>
            <xs:enumeration value="24"/>
            <xs:enumeration value="25"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="JobOutputOptions" type="JobOutputOptionsType" minOccurs="0"/>
      <xs:element name="ModelOffset" type="ModelOffsetType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:

- **MasterName** -- obsolete -- left in backwards compatibility - do not modify
- **MaxLines**
  - Set from 100 to 950.
  - Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

**Example use:**
```
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="900"
CallSub="M10"
```

**CallSub**
Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

**Example use:**
```
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="750"
CallSub="M10"
```

**ReturnSub**
Creates a RET statement in the current Robot Job when a specific M or G code is

**Example use:**
```
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="750"
CallSub="M11"
```
<table>
<thead>
<tr>
<th>type</th>
<th>InformFileFormatsType</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>content complex</td>
</tr>
<tr>
<td>attributes</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>MasterandSubs</td>
</tr>
<tr>
<td>Type</td>
<td>xs:string</td>
</tr>
<tr>
<td>Use</td>
<td>required</td>
</tr>
<tr>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>Fixed</td>
<td></td>
</tr>
<tr>
<td>annotation</td>
<td>documentation</td>
</tr>
<tr>
<td>This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:</td>
<td></td>
</tr>
<tr>
<td>MasterName</td>
<td>-- obsolete – left in backwards compatibility - do not modify</td>
</tr>
<tr>
<td>MaxLines</td>
<td>Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.</td>
</tr>
<tr>
<td>Example use:</td>
<td>InformFileFormats MasterandSubs=&quot;False&quot; MasterName=&quot;gmaster&quot; MaxLines=&quot;900&quot; CallSub=&quot;M10&quot;/</td>
</tr>
<tr>
<td>CallSub</td>
<td>Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This</td>
</tr>
</tbody>
</table>
function can be used for a number of functions.

Example use:
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="750"
CallSub="M10"/

ReturnSub
Creates a RET statement in the current Robot Job when a specific M or G code is

Example use:
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="750"
CallSub="M11"/

<table>
<thead>
<tr>
<th>MasterName</th>
<th>xs:string</th>
<th>optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxLines</td>
<td>xs:short</td>
<td>required</td>
</tr>
<tr>
<td>CallSub</td>
<td>xs:string</td>
<td>optional</td>
</tr>
<tr>
<td>ReturnSub</td>
<td>xs:string</td>
<td>optional</td>
</tr>
</tbody>
</table>

source <xs:element name="InformFileFormats" type="InformFileFormatsType"/>

element Root/Settings/ToolNumber

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:int</th>
</tr>
</thead>
</table>

properties

<table>
<thead>
<tr>
<th>isRef</th>
<th>content</th>
<th>default</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>simple</td>
<td>0</td>
</tr>
</tbody>
</table>
<xs:element name="ToolNumber" default="0">
  <xs:simpleType>
    <xs:restriction base="xs:int">
      <xs:enumeration value="0"/>
      <xs:enumeration value="1"/>
      <xs:enumeration value="2"/>
      <xs:enumeration value="3"/>
      <xs:enumeration value="4"/>
      <xs:enumeration value="5"/>
      <xs:enumeration value="6"/>
      <xs:enumeration value="7"/>
      <xs:enumeration value="8"/>
      <xs:enumeration value="9"/>
      <xs:enumeration value="10"/>
      <xs:enumeration value="11"/>
      <xs:enumeration value="12"/>
      <xs:enumeration value="13"/>
      <xs:enumeration value="14"/>
      <xs:enumeration value="15"/>
      <xs:enumeration value="16"/>
      <xs:enumeration value="17"/>
      <xs:enumeration value="18"/>
      <xs:enumeration value="19"/>
      <xs:enumeration value="20"/>
      <xs:enumeration value="21"/>
      <xs:enumeration value="22"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
element **Root/Settings/JobOutputOptions**

- **type**: `JobOutputOptionsType`
- **properties**:
  - `isRef 0`
  - `minOcc 0`
  - `maxOcc 1`
  - `content complex`
- **children**: `JobType`
- **source**: `<xs:element name="JobOutputOptions" type="JobOutputOptionsType" minOccurs="0"/>`

---

element **Root/Settings/ModelOffset**

- **type**: `ModelOffsetType`
| properties | isRef 0  
| minOcc 0  
| maxOcc 1  
| content complex |

| children | ModelOffsetX ModelOffsetY ModelOffsetZ ModelAngleRx ModelAngleRy ModelAngleRz |

| source | <xs:element name="ModelOffset" type="ModelOffsetType" minOccurs="0"/> |

element Rx

| diagram | ![Diagram Rx] |
| type | xs:double |
| properties | content simple |
| used by | element ThreeDOF |
| source | <xs:element name="Rx" type="xs:double"/> |

element Ry

| diagram | ![Diagram Ry] |
| type | xs:double |
| properties | content simple |
| used by | element ThreeDOF |
| source | <xs:element name="Ry" type="xs:double"/> |

element Rz

| diagram | ![Diagram Rz] |
| type | xs:double |
| properties | content simple |
| used by | element ThreeDOF |
| source | <xs:element name="Rz" type="xs:double"/> |

element ThreeDOF
Use this values to set the last length of the tool to be perpendicular to the part coordinate system. Generally this will be 180, 0, X Where X will be -90, 0, 90, 180, 270 degrees.

```
<xs:element name="ThreeDOF">
    <xs:annotation>
        <xs:documentation>Use this values to set the last length of the tool to be perpendicular to the part coordinate system. Generally this will be 180, 0, X Where X will be -90, 0, 90, 180, 270 degrees. </xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="Rx"/>
            <xs:element ref="Ry"/>
            <xs:element ref="Rz"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```
element `Units`

**Diagram**

```
    Units
     ├── LengthUnits
     │    └── Length
     ├── AngleUnits
     │    └── Degrees
     └── TimeUnits
          └── Minutes
```

- **Properties**
  - Content: complex
- **Children**
  - `LengthUnits`, `AngleUnits`, `TimeUnits`
- **Used by**
  - Element `Root/Settings`
- **Annotation**
  - Documentation:
    - This element, and its sub elements specify the Length, Angle, and Time Units used in this XML File.

**Source**

```xml
<xs:element name="Units">
  <xs:annotation>
    <xs:documentation>
      This element, and its sub elements specify the Length, Angle, and Time Units used in this XML File.
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="LengthUnits"/>
      <xs:element ref="AngleUnits"/>
      <xs:element ref="TimeUnits"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```
**attributes**

**MasterandSubs**

This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:

- **MasterName** -- obsolete -- left in backwards compatibility - do not modify

- **MaxLines**
  - Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

  Example use:
  ```
  InformFileFormats
  MasterandSubs="False"
  MasterName="gmaster"
  MaxLines="900"
  CallSub="M10"/
  ```

- **CallSub**
  - Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

  Example use:
  ```
  InformFileFormats
  MasterandSubs="False"
  MasterName="gmaster"
  MaxLines="750"
  CallSub="M10"/
  ```

- **ReturnSub**
  - Creates a RET statement in the current Robot Job when a specific M or G code is

  Example use:
  ```
  InformFileFormats
  MasterandSubs="False"
  MasterName="gmaster"
  MaxLines="750"
  CallSub="M11"/
  ```

- **InformFileFormats**

  This element controls the form of the robot jobs that are created. The following Attributes are used:

  - **MasterName** -- obsolete -- left in backwards compatibility - do not modify

  - **MaxLines**
    - Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

  Example use:  
  ```
  InformFileFormats
  MasterandSubs="False"
  MasterName="gmaster"
  MaxLines="900"
  CallSub="M10"/
  ```

- **CallSub**
  - Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

  Example use:  
  ```
  InformFileFormats
  MasterandSubs="False"
  MasterName="gmaster"
  MaxLines="750"
  CallSub="M10"/
  ```

- **MasterName**

  - Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.
This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MasterSubs</td>
<td>xs:string</td>
<td>required</td>
<td></td>
<td></td>
<td>This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:</td>
</tr>
<tr>
<td>MaxLines</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MasterName -- obsolete – left in backwards compatibility - do not modify</td>
</tr>
</tbody>
</table>
| CallSub      |           |       |         |       | MaxLines
| ReturnSub    |           |       |         |       | Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created. |

Example use:
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="750" CallSub="M10"/

Example use: InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="750" CallSub="M11"/

CallSub
Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used
for a number of functions.

Example use:
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="750"
CallSub="M10"

ReturnSub
Creates a RET statement in the current Robot Job when a specific M or G code is

Example use:
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="750"
CallSub="M11"

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MasterName</td>
<td>xs:string</td>
<td>optional</td>
</tr>
<tr>
<td>MaxLines</td>
<td>xs:short</td>
<td>required</td>
</tr>
<tr>
<td>CallSub</td>
<td>xs:string</td>
<td>optional</td>
</tr>
<tr>
<td>ReturnSub</td>
<td>xs:string</td>
<td>optional</td>
</tr>
</tbody>
</table>

This element controls the form of the robot jobs that are created. The following Attributes are used:

MasterName -- obsolete – left in backwards compatibility - do not modify

MaxLines
Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="900" CallSub="M10"

CallSub
Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M10"

ReturnSub
Creates a RET statement in the current Robot Job when a specific M or G code is

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M11"
<xs:complexType name="InformFileFormatsType">
  <xs:annotation>
    <xs:documentation>This element controls the form of the robot jobs that are created. The following Attributes are used:

MasterName -- obsolete – left in backwards compatibility - do not modify

MaxLines
Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="900" CallSub="M10"/

CallSub
Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M10"/

ReturnSub
Creates a RET statement in the current Robot Job when a specific M or G code is found.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M11"/
</xs:documentation>
  </xs:annotation>
  <xs:attribute name="MasterandSubs" type="xs:string" use="required">
    <xs:annotation>
      <xs:documentation>This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:

MasterName -- obsolete – left in backwards compatibility - do not modify

MaxLines
Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="900" CallSub="M10"/

CallSub
Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M11"/
</xs:documentation>
  </xs:annotation>
</xs:complexType>
MaxLines="750" CallSub="M10"/

ReturnSub
Creates a RET statement in the current Robot Job when a specific M or G code is

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster"
MaxLines="750" CallSub="M11"/

attribute InformFileFormatsType/@MasterandSubs

<table>
<thead>
<tr>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
</table>
| properties   | isRef 0
use required  |

annotation documentation
This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:

MasterName -- obsolete – left in backwards compatibility - do not modify

MaxLines
Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="900" CallSub="M10"/

CallSub
Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M10"/

ReturnSub
Creates a RET statement in the current Robot Job when a specific M or G code is

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M11"/
This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:

**MasterName** -- obsolete – left in backwards compatibility - do not modify

**MaxLines**
Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="900" CallSub="M10"/

**CallSub**
Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M10"/

**ReturnSub**
Creates a RET statement in the current Robot Job when a specific M or G code is found. This function can be used for a number of functions.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M11"/

<table>
<thead>
<tr>
<th>attribute</th>
<th>InformFileFormatsType/@MasterName</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>xs:string</td>
</tr>
<tr>
<td>properties</td>
<td>useRef 0 use optional</td>
</tr>
<tr>
<td>source</td>
<td><code>&lt;xs:attribute name=&quot;#MasterName&quot; type=&quot;xs:string&quot; use=&quot;optional&quot;/&gt;</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>attribute</th>
<th>InformFileFormatsType/@MaxLines</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>xs:short</td>
</tr>
<tr>
<td>properties</td>
<td>useRef 0 use required</td>
</tr>
<tr>
<td>source</td>
<td><code>&lt;xs:attribute name=&quot;#MaxLines&quot; type=&quot;xs:short&quot; use=&quot;required&quot;/&gt;</code></td>
</tr>
</tbody>
</table>
attribute `InformFileFormatsType/@CallSub`  
<table>
<thead>
<tr>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>use optional</td>
</tr>
<tr>
<td>source</td>
<td><code>&lt;xs:attribute name=&quot;CallSub&quot; type=&quot;xs:string&quot; use=&quot;optional&quot;/&gt;</code></td>
</tr>
</tbody>
</table>

attribute `InformFileFormatsType/@ReturnSub`  
<table>
<thead>
<tr>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>use optional</td>
</tr>
<tr>
<td>source</td>
<td><code>&lt;xs:attribute name=&quot;ReturnSub&quot; type=&quot;xs:string&quot; use=&quot;optional&quot;/&gt;</code></td>
</tr>
</tbody>
</table>

complexType `JobOutputOptionsType`

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Diagram]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>children</th>
<th>JobType</th>
</tr>
</thead>
</table>

| used by | element `Root/Settings/JobOutputOptions` |

<table>
<thead>
<tr>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;xs:complexType name=&quot;JobOutputOptionsType&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:all&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element name=&quot;JobType&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:complexType&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:choice&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element name=&quot;PulseJobType&quot; default=&quot;True&quot; minOccurs=&quot;0&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:simpleType&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:restriction base=&quot;xs:string&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:enumeration value=&quot;True&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:enumeration value=&quot;False&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:restriction&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:complexType&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element name=&quot;RectanJobType&quot; minOccurs=&quot;0&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:complexType&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:sequence&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element name=&quot;PositionConfiguration&quot; type=&quot;PositionConfigurationType&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element name=&quot;RobotPosture&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:complexType&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:all&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element name=&quot;FlipPosture&quot; default=&quot;0&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:simpleType&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:restriction base=&quot;xs:integer&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:enumeration value=&quot;0&quot;/&gt;</code></td>
</tr>
</tbody>
</table>
<xs:element name="UpperPosture" default="0">
    <xs:simpleType>
        <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>

<xs:element name="FrontPosture" default="0">
    <xs:simpleType>
        <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>

<xs:element name="RPosture" default="0">
    <xs:simpleType>
        <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>

<xs:element name="TPosture" default="0">
    <xs:simpleType>
        <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
**element** JobOutputOptionsType/JobType

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="Diagram of JobOutputOptionsType/JobType" /></th>
</tr>
</thead>
</table>
| properties | isRef 0  
content complex |
| children | PulseJobType RectanJobType |

**source**

```xml
<xs:element name="JobType">
  <xs:complexType>
    <xs:choice>
      <xs:element name="PulseJobType" default="True" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="True"/>  
            <xs:enumeration value="False"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="RectanJobType" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="PositionConfiguration" type="PositionConfigurationType"/>
            <xs:element name="RobotPosture">
              <xs:complexType>
                <xs:all>
                  <xs:element name="FlipPosture" default="0">
                    <xs:simpleType>
                      <xs:restriction base="xs:integer">
                        <xs:enumeration value="0"/>  
                        <xs:enumeration value="1"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                  <xs:element name="UpperPosture" default="0">
                    <xs:simpleType>
                      <xs:restriction base="xs:integer">
                        <xs:enumeration value="0"/>  
                        <xs:enumeration value="1"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                  <xs:element name="FrontPosture" default="0">
                    <xs:simpleType>
                      <xs:restriction base="xs:integer">
                        <xs:enumeration value="0"/>  
                        <xs:enumeration value="1"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                </xs:all>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>
</xs:element>
```
<xs:element name="RPosture" default="0">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:enumeration value="0"/>
      <xs:enumeration value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

<xs:element name="TPosture" default="0">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:enumeration value="0"/>
      <xs:enumeration value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

---

**element** JobOutputOptionsType/JobType/PulseJobType

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="PulseJobType" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of <strong>xs:string</strong></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>properties</th>
<th>isRef 0</th>
<th>minOcc 0</th>
<th>maxOcc 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>content</td>
<td>simple</td>
<td></td>
<td></td>
</tr>
<tr>
<td>default</td>
<td>True</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| facets     | enumeration True | enumeration False |</p>
<table>
<thead>
<tr>
<th>source</th>
<th><code>&lt;xs:element name=&quot;PulseJobType&quot; default=&quot;True&quot; minOccurs=&quot;0&quot;&gt;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>&lt;xs:simpleType&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:restriction base=&quot;xs:string&quot;&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;True&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;False&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/xs:restriction&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/xs:simpleType&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/xs:element&gt;</code></td>
</tr>
</tbody>
</table>

**element JobOutputOptionsType/JobType/RectanJobType**

**Diagram**

```
RectanJobType [ ]
    ---- [ ]
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
    |     |
element JobOutputOptionsType/JobType/RectanJobType/PositionConfiguration

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>PositionConfigurationType</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td>content</td>
<td>complex</td>
</tr>
<tr>
<td>children</td>
<td>RobotFrame BaseFrame UserFrame</td>
</tr>
<tr>
<td>source</td>
<td><code>&lt;xs:element name=&quot;PositionConfiguration&quot; type=&quot;PositionConfigurationType&quot;/&gt;</code></td>
</tr>
</tbody>
</table>
<xs:element name="RobotPosture">
  <xs:complexType>
    <xs:all>
      <xs:element name="FlipPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="UpperPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="FrontPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="RPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="TPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:all>
  </xs:complexType>
</xs:element>
element JobOutputOptionsType/JobType/RectanJobType/RobotPosture/FlipPosture

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>content simple</td>
</tr>
<tr>
<td></td>
<td>default 0</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration 0</td>
</tr>
<tr>
<td></td>
<td>enumeration 1</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;FlipPosture&quot; default=&quot;0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:restriction base=&quot;xs:integer&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;1&quot;/&gt;</td>
</tr>
</tbody>
</table>

element JobOutputOptionsType/JobType/RectanJobType/RobotPosture/UpperPosture

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>content simple</td>
</tr>
<tr>
<td></td>
<td>default 0</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;UpperPosture&quot; default=&quot;0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:restriction base=&quot;xs:integer&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;1&quot;/&gt;</td>
</tr>
</tbody>
</table>


| facets | enumeration 0  
|        | enumeration 1  |

| source | `<xs:element name="UpperPosture" default="0">`  
|        | `<xs:simpleType>`  
|        | `<xs:restriction base="xs:integer">`  
|        | `<xs:enumeration value="0"/>`  
|        | `<xs:enumeration value="1"/>`  
|        | `</xs:restriction>`  
|        | `</xs:simpleType>`  
|        | `</xs:element>` |

**element JobOutputOptionsType/JobType/RectanJobType/RobotPosture/FrontPosture**

| diagram | ![Diagram](link) |
| type    | restriction of `xs:integer` |
| properties | isRef 0  
|          | content simple  
|          | default 0  
| facets   | enumeration 0  
|          | enumeration 1  

| source | `<xs:element name="FrontPosture" default="0">`  
|        | `<xs:simpleType>`  
|        | `<xs:restriction base="xs:integer">`  
|        | `<xs:enumeration value="0"/>`  
|        | `<xs:enumeration value="1"/>`  
|        | `</xs:restriction>`  
|        | `</xs:simpleType>`  
|        | `</xs:element>` |

**element JobOutputOptionsType/JobType/RectanJobType/RobotPosture/RPosture**

| diagram | ![Diagram](link) |
| type    | restriction of `xs:integer` |
| properties | isRef 0  
|          | content simple  
|          | default 0  
| facets   | enumeration 0  
|          | enumeration 1  |
element JobOutputOptionsType/JobType/RectanJobType/RobotPosture/TPosture

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:integer</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>properties</th>
<th>isRef 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>content</td>
<td>simple</td>
</tr>
<tr>
<td>default</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>facets</th>
<th>enumeration 0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>enumeration 1</td>
</tr>
</tbody>
</table>

source

<xs:element name="RPosture" default="0">
    <xs:simpleType>
        <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>

complexType ModelOffsetType

diagram

<table>
<thead>
<tr>
<th>children</th>
<th>ModelOffsetX ModelOffsetY ModelOffsetZ ModelAngleRx ModelAngleRy ModelAngleRz</th>
</tr>
</thead>
</table>
used by element Root/Settings/ModelOffset

<table>
<thead>
<tr>
<th>source</th>
</tr>
</thead>
</table>
| <xs:complexType name="ModelOffsetType">  
  <xs:sequence>
    <xs:element name="ModelOffsetX" type="xs:double" default="0.0" minOccurs="0"/>
    <xs:element name="ModelOffsetY" type="xs:double" default="0.0" minOccurs="0"/>
    <xs:element name="ModelOffsetZ" type="xs:double" default="0.0" minOccurs="0"/>
    <xs:element name="ModelAngleRx" type="xs:double" default="0.0" minOccurs="0"/>
    <xs:element name="ModelAngleRy" type="xs:double" default="0.0" minOccurs="0"/>
    <xs:element name="ModelAngleRz" type="xs:double" default="0.0" minOccurs="0"/>
  </xs:sequence>  
</xs:complexType> |

element ModelOffsetType/ModelOffsetX

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>xs:double</th>
</tr>
</thead>
</table>

properties

- isRef: 0
- minOcc: 0
- maxOcc: 1
- content: simple
- default: 0.0

source

<x:element name="ModelOffsetX" type="xs:double" default="0.0" minOccurs="0"/>

element ModelOffsetType/ModelOffsetY

| type | xs:double |

properties

- isRef: 0
- minOcc: 0
- maxOcc: 1
- content: simple
- default: 0.0

source

<x:element name="ModelOffsetY" type="xs:double" default="0.0" minOccurs="0"/>

element ModelOffsetType/ModelOffsetZ

diagram

| type | xs:double |

properties

- isRef: 0
- minOcc: 0
- maxOcc: 1
- content: simple
- default: 0.0

source

<x:element name="ModelOffsetZ" type="xs:double" default="0.0" minOccurs="0"/>
<xs:element name="ModelOffsetZ" type="xs:double" default="0.0" minOccurs="0"/>

element ModelOffsetType/ModelAngleRx

diagram

modelAngleRx

type xs:double

<xs:element name="ModelAngleRx" type="xs:double" default="0.0" minOccurs="0"/>

element ModelOffsetType/ModelAngleRy

diagram

modelAngleRy

type xs:double

<xs:element name="ModelAngleRy" type="xs:double" default="0.0" minOccurs="0"/>

element ModelOffsetType/ModelAngleRz

diagram

modelAngleRz

type xs:double

<xs:element name="ModelAngleRz" type="xs:double" default="0.0" minOccurs="0"/>
**complexType PositionConfigurationType**

<table>
<thead>
<tr>
<th>Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
</tr>
</tbody>
</table>

**children**
- RobotFrame
- BaseFrame
- UserFrame

**used by**
- element JobOutputOptionsType/JobType/RectanJobType/PositionConfiguration

**source**
```
<xs:complexType name="PositionConfigurationType">
    <xs:choice>
        <xs:element name="RobotFrame" type="xs:string" default="True" minOccurs="0"/>
        <xs:element name="BaseFrame" default="False" minOccurs="0">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="True"/>
                    <xs:enumeration value="False"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="UserFrame" default="1" minOccurs="0">
            <xs:simpleType>
                <xs:restriction base="xs:integer">
                    <xs:totalDigits value="2"/>
                    <xs:fractionDigits value="0"/>
                    <xs:minInclusive value="0"/>
                    <xs:maxInclusive value="25"/>
                    <xs:pattern value="\d{1,2}"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
    </xs:choice>
</xs:complexType>
```

**element PositionConfigurationType/RobotFrame**

<table>
<thead>
<tr>
<th>Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
</tr>
</tbody>
</table>

**type**
- xs:string
### PositionConfigurationType/BaseFrame

**Diagram:**

```
BaseFrame
```

**Type:** restriction of xs:string

**Properties:**

- isRef: 0
- minOccurs: 0
- maxOccurs: 1
- content: simple
- default: False

**Facets:**

- enumeration True
- enumeration False

**Source:**

```xml
<xs:element name="BaseFrame" default="False" minOccurs="0"/>
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:enumeration value="True"/>
    <xs:enumeration value="False"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
```

### PositionConfigurationType/UserFrame

**Diagram:**

```
UserFrame
```

**Type:** restriction of xs:integer

**Properties:**

- isRef: 0
- minOccurs: 0
- maxOccurs: 1
- content: simple
- default: 1

**Facets:**

- minInclusive: 0
- maxInclusive: 25
- totalDigits: 2
- fractionDigits: 0
- pattern: \d{1,2}
<xs:element name="UserFrame" default="1" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:totalDigits value="2"/>
      <xs:fractionDigits value="0"/>
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="25"/>
      <xs:pattern value="\d{1,2}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
### Elements Complex types

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>B</strong></td>
<td><strong>ArmConfigType</strong></td>
</tr>
<tr>
<td><strong>L</strong></td>
<td></td>
</tr>
<tr>
<td><strong>R</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Root</strong></td>
<td></td>
</tr>
<tr>
<td><strong>S</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Settings</strong></td>
<td></td>
</tr>
<tr>
<td><strong>T</strong></td>
<td></td>
</tr>
<tr>
<td><strong>U</strong></td>
<td></td>
</tr>
</tbody>
</table>

#### element B

<table>
<thead>
<tr>
<th><strong>Diagram</strong></th>
<th><img src="image" alt="Diagram B" /></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>restriction of xs:short</td>
</tr>
<tr>
<td><strong>Properties</strong></td>
<td>content simple</td>
</tr>
<tr>
<td><strong>Used by</strong></td>
<td>complexType <strong>ArmConfigType</strong></td>
</tr>
<tr>
<td><strong>Facets</strong></td>
<td>enumeration 26546</td>
</tr>
</tbody>
</table>
| **Source**  | `<xs:element name="B">  
  <xs:simpleType>  
    <xs:restriction base="xs:short">  
      <xs:enumeration value="26546"/>  
    </xs:restriction>  
  </xs:simpleType>  
</xs:element>` |

#### element L

<table>
<thead>
<tr>
<th><strong>Diagram</strong></th>
<th><img src="image" alt="Diagram L" /></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>restriction of xs:short</td>
</tr>
<tr>
<td><strong>Properties</strong></td>
<td>content simple</td>
</tr>
<tr>
<td><strong>Used by</strong></td>
<td>complexType <strong>ArmConfigType</strong></td>
</tr>
<tr>
<td><strong>Facets</strong></td>
<td>enumeration -16903</td>
</tr>
</tbody>
</table>
**element R**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="Diagram" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:byte</td>
</tr>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>complexType ArmConfigType</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration 0</td>
</tr>
</tbody>
</table>

**source**

```xml
<xs:element name="R">
  <xs:simpleType>
    <xs:restriction base="xs:byte">
      <xs:enumeration value="0"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**element Root**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="Diagram" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content complex</td>
</tr>
<tr>
<td>children</td>
<td>Settings</td>
</tr>
</tbody>
</table>

**source**

```xml
<xs:element name="Root">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Settings"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**element S**
### element Settings

**Diagram:**

```
\[ \text{Settings} \rightarrow \text{ArmConfig} \]
```

**Properties:**
- content: complex

**Children:**
- ArmConfig

**Used by:**
- element Root

**Source:**
```
<xs:element name="Settings">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ArmConfig" type="ArmConfigType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

### element Settings/ArmConfig

---

```xml
<xs:element name="S">
  <xs:simpleType>
    <xs:restriction base="xs:short">
      <xs:enumeration value="24728"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```
```
<xs:element name="ArmConfig" type="ArmConfigType"/>
```
### element U

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:int</td>
</tr>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>complexType ArmConfigType</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration -52854</td>
</tr>
</tbody>
</table>

```xml
<xs:element name="U">
  <xs:simpleType>
    <xs:restriction base="xs:int">
      <xs:enumeration value="-52854"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

### complexType ArmConfigType

| diagram | ![](image) |

```xml
<xs:simpleType>
  <xs:restriction base="xs:int">
    <xs:enumeration value="-56400"/>
  </xs:restriction>
</xs:simpleType>
```
children: **S L U R B T**

used by: element `Settings/ArmConfig`

attributes:
- **JobName** `xs:string` *optional*

annotation documentation:

Specifies the robot initial starting position and form. This position is specified in joint coordinates and will set the initial pose (elbow up/dow, etc.) of the robot.

This position can be set by moving the simulated robot to the part coordinate system 0,0,0 location and setting the individual axes to archive the desired pose. The values can be copied from MotoSimEG and pasted in the appropriate joint location.

Note: You can specify up to 21 unique robot configurations.

Note: The first set of Arm Configuration values are automatically selected.
<xs:complexType name="ArmConfigType">
  <xs:annotation>
    <xs:documentation>
      Specifies the robot initial starting position and form. This position is specified in joint coordinates and will set the initial pose (elbow up/dow, etc.) of the robot.
    </xs:documentation>
    <xs:documentation>
      This position can be set by moving the simulated robot to the part coordinate system 0,0,0 location and setting the individual axes to archive the desired pose. The values can be copied from MotoSimEG and pasted in the appropriate joint location.
    </xs:documentation>
    <xs:documentation>
      Note: You can specify up to 21 unique robot configurations.
    </xs:documentation>
    <xs:documentation>
      Note: The first set of Arm Configuration values are automatically selected.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element ref="S"/>
    <xs:element ref="L"/>
    <xs:element ref="U"/>
    <xs:element ref="R"/>
    <xs:element ref="B"/>
    <xs:element ref="T"/>
  </xs:sequence>
  <xs:attribute name="JobName" type="xs:string" use="optional"/>
</xs:complexType>

<table>
<thead>
<tr>
<th>attribute</th>
<th>ArmConfigType/@JobName</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>xs:string</td>
</tr>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>use optional</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:attribute name=&quot;JobName&quot; type=&quot;xs:string&quot; use=&quot;optional&quot;/&gt;</td>
</tr>
</tbody>
</table>
### Schema \textit{GCodeSchemaV10.xsd}

**schema location:** C:\Program Files\Motoman\MotoSim EG\Schemas-GCode\GCodeSchemaV10.xsd

**attribute form default:** qualified

<table>
<thead>
<tr>
<th>Elements</th>
<th>Complex types</th>
<th>Simple types</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlignXwithPath</td>
<td>ArmConfigType</td>
<td>ReferenceTypeBase</td>
</tr>
<tr>
<td>AngleSystem</td>
<td>BaseStationType</td>
<td></td>
</tr>
<tr>
<td>AngleUnits</td>
<td>DefaultVelocityType</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>ExternalAxesType</td>
<td></td>
</tr>
<tr>
<td>CallJob</td>
<td>FileConfigurationType</td>
<td></td>
</tr>
<tr>
<td>CannedVelocity</td>
<td>FourToSixAxisType</td>
<td></td>
</tr>
<tr>
<td>CircleFixUp</td>
<td>GCODE</td>
<td></td>
</tr>
<tr>
<td>Const</td>
<td>GCodeToInform</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>INFORM</td>
<td></td>
</tr>
<tr>
<td>DestIndex</td>
<td>InformFileFormatsType</td>
<td></td>
</tr>
<tr>
<td>DestNum</td>
<td>InformMotionFunctionGroupType</td>
<td></td>
</tr>
<tr>
<td>DestType</td>
<td>InputPositionDataType</td>
<td></td>
</tr>
<tr>
<td>DistVar</td>
<td>JobOutputOptionsType</td>
<td></td>
</tr>
<tr>
<td>EndCannedCycle</td>
<td>MotionFunctionsType</td>
<td></td>
</tr>
<tr>
<td>GCODE</td>
<td>ParseGroupsThrowAwayLinesType</td>
<td></td>
</tr>
<tr>
<td>GenInform</td>
<td>ParseGroupsType</td>
<td></td>
</tr>
<tr>
<td>GTOINFORM</td>
<td>PositionConfigurationPulseType</td>
<td></td>
</tr>
<tr>
<td>Header</td>
<td>PositionConfigurationType</td>
<td></td>
</tr>
<tr>
<td>INF</td>
<td>ReferenceModelType</td>
<td></td>
</tr>
<tr>
<td>INFORMATION</td>
<td>ReferenceModelType</td>
<td></td>
</tr>
<tr>
<td>InformMotionFunctionGroup</td>
<td>RotatingStationJobDataPositionType</td>
<td></td>
</tr>
<tr>
<td>InputPositionData</td>
<td>RotatingStationSetupType</td>
<td></td>
</tr>
<tr>
<td>isLiteralInform</td>
<td>RotatingStationToCNCAngleMappingType</td>
<td></td>
</tr>
<tr>
<td>JobOutputOptions</td>
<td>Search</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>ShiftOffset</td>
<td></td>
</tr>
<tr>
<td>LengthUnits</td>
<td>ThreeDOFType</td>
<td></td>
</tr>
<tr>
<td>MotionFunction</td>
<td>UnitsType</td>
<td></td>
</tr>
<tr>
<td>MotionFunctionList</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MotionGroup</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MotionType</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MoveTag</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OutputOff</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OutputOn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PositionType</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PosLevel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ReferenceModel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RefFrame</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RegExParseInt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Root</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetPosElmnt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetVar</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ShiftFunctionType</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ShiftOffset</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ShiftOffsetDVarIndex</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ShiftOffsetTrueFalse</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ShiftOn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ShiftReferenceModel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ShiftSourceNum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SourceNum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StartCannedCycle</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
element \texttt{AlignXwithPath}

| diagram | 
| --- | --- |
| | 

| type | restriction of \texttt{xs:string} |
| properties | content \texttt{simple} default \texttt{False} |
| use by | complexType \texttt{FileConfigurationType} |
| facets | enumeration \texttt{True} enumeration \texttt{False} |
| annotation | documentation \texttt{Reserved for Future Development.} |
| source | \begin{verbatim}
<xs:element name="AlignXwithPath" default="False">
  <xs:annotation>
    <xs:documentation>Reserved for Future Development.</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="True"/>
      <xs:enumeration value="False"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
\end{verbatim} |

element \texttt{AngleSystem}
The AngleSystem element is a required element that defines the angle system used in the G Code file. Values are YPR (Yaw, Pitch, and Roll) or Euler.

YPR is the Motoman and MotoSimEg standard

Euler angles is method of defining three rotations of an object about a point. It is somewhat of a standard in the Aerospace and Graphics industry. Euler Angles generally rotate about the ZYZ axes.

Example use: `<AngleSystem>YPR</AngleSystem>`
<!-- The AngleSystem element is a required element that defines the angle system used in the G Code file. Values are YPR (Yaw, Pitch, and Roll) or Euler. -->
</xs:element>

**element AngleUnits**

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
</table>

| type       | restriction of xs:string |
| properties | content simple            |
| used by    | complexType UnitsType    |
| facets     | enumeration Degrees      |
|            | enumeration Radians      |
| source     | `<xs:element name="AngleUnits">
|           |   `<xs:simpleType>
|           |     `<xs:restriction base="xs:string">
|           |       `<xs:enumeration value="Degrees"/>
|           |       `<xs:enumeration value="Radians"/>
|           |     </xs:restriction>
|           |   </xs:simpleType>
|           | </xs:element>` |

**element B**

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
</table>

| type       | xs:integer |
| properties | content simple |
| used by    | complexType ArmConfigType |
| source     | `<xs:element name="B" type="xs:integer"/>` |

**element CallJob**

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
</table>
This is an XML to Inform Instruction. Use this element to cause the robot to call a job subroutine.

This instruction will appear in the intermediate XML file that the G Code Converter creates during the conversion process. It will produce an Inform Instruction in the robot jbi with

Example use:  <CallJob>Pickup</CallJob>
Example output JBI: CALL JOB:Pickup
type restriction of xs:integer

properties content simple

facets enumeration 1
enumeration 2
enumeration 3
enumeration 4
enumeration 5
enumeration 6
enumeration 7
enumeration 8
enumeration 9

annotation documentation
Used with the CannedVelocityRegister Attribute in the InformMotionFunctionGroup Command

source
<xs:element name=\"CannedVelocity\">
  <xs:annotation>
    <xs:documentation>Used with the CannedVelocityRegister Attribute in the InformMotionFunctionGroup Command</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:enumeration value="1"/>
      <xs:enumeration value="2"/>
      <xs:enumeration value="3"/>
      <xs:enumeration value="4"/>
      <xs:enumeration value="5"/>
      <xs:enumeration value="6"/>
      <xs:enumeration value="7"/>
      <xs:enumeration value="8"/>
      <xs:enumeration value="9"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

element CircleFixUp

diagram

Machine tools and Motoman robots produce circles in very different ways.

G Code Circle Overview

G02: Circular motion, clockwise direction. There are two circular motion commands for cutting arcs, which are segments of circles, and full circles. Circular motion commands always move at the feedrate and are not rapid motion commands. The G02 command is the instruction for a clockwise circular motion. The arc will start at the tools current location and sweep around clockwise to the designated ending location at the
clockwise to the designated ending location at the programmed feedrate. There are several 'standard' methods to define the radius of the arc. These methods are discussed below, after the discussion on the counter-clockwise command.

G03: Circular motion, counter-clockwise direction. The G03 command is the instruction for a counter-clockwise circular motion. The arc will start at the tool's current location and sweep around counter-clockwise to the designated ending location at the programmed feedrate. There are several 'standard' methods to define the radius of an arc. These methods are discussed below, after the discussion on the counter-clockwise command.

Circular motion format. Arc radius definitions: A standard circular motion G-code instruction will always designate the arc's ending point in X, Y, and Z. However, there are several 'standard' formats for programming the arc's radius. Some controls support only one format, others support two or more formats. Refer to your machine's programming manual to determine which formats your CNC control supports. The simplest method supported by many controls is to just include the radius value in the G-code line. The CNC control will do the calculations from there! For controls that support this format, the radius value is programmed by an 'R' word. In the following example a 0.5 inch radius will be machined (The Z value is modal in this example). Note: If the CNC control can't calculate the defined radius, an error will occur at the machine.

G02 X1.0 Y1.0 R0.5 F10. The most commonly supported format is the usage of I and J words to define the arc's center point location. Unfortunately, there are several different formats for defining the I and J locations! The first I and J format some controls support, one that is pretty straightforward, is to program the arc center-point in absolute coordinates.
Machine tools and Motoman robots produce circles in very different ways.

G Code Circle Overview

G02: Circular motion, clockwise direction. There are two circular motion commands for cutting arcs, which are segments of circles, and full circles. Circular motion commands always move at the feedrate and are not rapid motion commands. The G02 command is the instruction for a clockwise circular motion. The arc will start at the tool's current location and sweep around clockwise to the designated ending location at the programmed feedrate.
There are several 'standard' methods to define the radius of the arc. These methods are discussed below, after the discussion on the counter-clockwise command.

Circular motion format, Arc Radius Definitions: A standard circular motion G-code instruction will always designate the arc's ending point in X, Y, and Z. However, there are several 'standard' formats for programming the arc's radius. Some controls support only one format, others support two or more formats. Refer to your machine's programming manual to determine which formats your CNC control supports.

The simplest method supported by many controls is to just include the radius value in the G-code line. The CNC control will do the calculations from there! For control's that support this format, the radius value is programmed by an 'R' word. In the following example a 0.5 inch radius will be machined (The Z value is modal in this example). Note: If the CNC control can't calculate the defined radius, an error will occur at the machine.
G02 X1.0 Y1.0 R0.5 F10.

The most commonly supported format is the usage of I and J words to define the arc's center point location. Unfortunately, there are several different formats for defining the I and J locations!
The first I and J format some controls support, one that is pretty straightforward, is to program the arc center-point in absolute coordinates. Program the arc's center-point location in the X axis with the 'I' word and the arc's center-point location in the Y axis with the 'J' word. This format, again one that makes a 0.5 inch radius, is written as follows:
G02 X1.75 Y1.25 I1.25 J0.75 F10.

G03: Circular motion, counter-clockwise direction. The G03 command is the instruction for a counter-clockwise circular motion. The arc will start at the tool's current location and sweep around counter-clockwise to the designated ending location at the programmed feedrate.
There are several 'standard' methods to define the radius of an arc. These methods are discussed below, after the discussion on the counter-clockwise command.
Machine tools and Motoman robots produce circles in very different ways.

G Code Circle Overview

G02: Circular motion, clockwise direction. There are two circular motion commands for cutting arcs, which are segments of circles, and full circles. Circular motion commands always move at the feedrate and are not rapid motion commands. The G02 command is the instruction for a clockwise circular motion. The arc will start at the tools current location and sweep around clockwise to the designated ending location at the programmed feedrate. There are several 'standard' methods to define the radius of the arc. These methods are discussed below, after the discussion on the counter-clockwise command.

G03: Circular motion, counter-clockwise direction. The G03 command is the instruction for a counter-clockwise circular motion. The arc will start at the tools current location and sweep around counter-clockwise to the designated ending location at the programmed feedrate. There are several 'standard' methods to define the radius of an arc. These methods are discussed below, after the discussion on the counter-clockwise command.

Circular motion format, Arc Radius Definitions: A standard circular motion G-code instruction will always designate the arc's ending point in X, Y, and Z. However, there are several 'standard' formats for programming the arc's radius. Some controls support only one format, others support two or more formats. Refer to your machine's programming manual to determine which formats your CNC control supports. The simplest method supported by many controls is to just include the radius value in the G-code line. The CNC control will do the calculations from there! For controls that support this format, the radius value is programmed by an 'R' word. In the following example a 0.5 inch radius will be machined (The Z value is modal in this example). Note: If the CNC control can't calculate the defined radius, an error will occur at the machine.

G02 X1.0 Y1.0 R0.5 F10.

The most commonly supported format is the usage of I and J words to define the arc's center point location. Unfortunately, there are several different formats for defining the I and J locations! The first I and J format some controls support, one that is pretty straightforward, is to program the arc center-point in absolute coordinates. Program the arc's center-point location in the X axis with the 'I' word and the arc's center-point location in the Y axis with the 'J' word. This format, again one that makes a 0.5 inch radius, is written as follows:

G02 X1.75 Y1.25 I1.25 J0.75 F10.
element `DestIndex`

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="DestIndex" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>xs:integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content: simple</td>
</tr>
<tr>
<td>used by</td>
<td>element: <code>SetPosElmnt</code></td>
</tr>
<tr>
<td>source</td>
<td><code>&lt;xs:element name=&quot;DestIndex&quot; type=&quot;xs:integer&quot; default=&quot;1&quot;/&gt;</code></td>
</tr>
</tbody>
</table>

element `DestNum`

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="DestNum" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>xs:integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content: simple</td>
</tr>
<tr>
<td>used by</td>
<td>element: <code>SetVar</code></td>
</tr>
<tr>
<td>source</td>
<td><code>&lt;xs:element name=&quot;DestNum&quot; type=&quot;xs:integer&quot;/&gt;</code></td>
</tr>
</tbody>
</table>
### Destination Number (DestNum)

- **Type**: `xs:integer`
- **Properties**: Content simple
- **Default**: 1
- **Used by Elements**: `SetPosElmnt`, `SetVar`

```xml
<xs:element name="DestNum" type="xs:integer" default="1"/>
```

### Destination Type (DestType)

- **Type**: `xs:string`
- **Properties**: Content simple
- **Default**: D
- **Used by Element**: `SetVar`

```xml
<xs:element name="DestType" type="xs:string" default="D"/>
```

### Distance Variable (DistVar)

- **Type**: `xs:anySimpleType`
- **Properties**: Content simple
- **Default**: 1
- **Used by ComplexType**: `Search`

```xml
<xs:element name="DistVar" type="xs:anySimpleType" default="1"/>
```

### End Canned Cycle (EndCannedCycle)

- **Type**: Restriction of `xs:string`
- **Properties**: Content simple
- **Used by ComplexType**: `GCODE`
- **Facets**: Enumeration True, Enumeration False
- **Annotation Documentation**: Ends the G Code Canned Cycle.
source <xs:element name="EndCannedCycle">
  <xs:annotation>
    <xs:documentation>Ends the G Code Canned Cycle.</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="True"/>
      <xs:enumeration value="False"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

element GCODE

This element is a sub-element of the G Code to Inform Instruction. It lists one G Code Instruction that will be converted into Inform.

This instruction is used in the G Code Tag. It indicates if the G Code function will map into an Inform Header Instruction. Values may be True or False.

This command is used to list all of the G Code functions that will cause robot motion. It is used by the G Code Converter during the G Code to Inform Conversion to parse all commands with multiple instructions and place the motion functions last, next to the XYZ position values.

When true, this command allows the XML commands in the corresponding Inform tag to be placed in the XML intermediate file without an further processing.

Used to turn on or Off a robot shift. Usually used with a G81 command.
type **GCODE**

properties content `complex`

children `Header` `Description` `MotionFunction` `isLiteralInform` `ShiftOffsetTrueFalse` `MotionGroup` `EndCannedCycle` `StartCannedCycle`

used by complexType **GCodeToInform**

attributes Name Type Use Default Fixed annotation

<table>
<thead>
<tr>
<th>attr</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>func</td>
<td></td>
<td>xs: anySimpleType</td>
<td>optional</td>
<td></td>
<td></td>
<td>This element is a sub-element of the G Code to Inform Instruction. It lists one G Code Instruction that will be converted into Inform.</td>
</tr>
</tbody>
</table>

source

```
<xs:element name="GCODE" type="GCODE">
  <xs:annotation>
    <xs:documentation>This element is a sub-element of the G Code to Inform Instruction. It lists one G Code Instruction that will be converted into Inform.</xs:documentation>
  </xs:annotation>
</xs:element>
```

**element GCodeToInform**

diagram
GenInform is an abbreviation for General Inform. Any valid Inform command can be used inside the GenInform Command but it is the responsibility of the user to know the correct syntax. This is a pass through command that is not syntax checked.

This command allows users to use any map an G Code Code command into any Inform Instruction. It is used in the GTOINFORM-- INFORM section of the Configuration File to map a G Code Instruction into a Robot Inform Instruction.

Example:
The following example maps a G90 Instruction into a Shift Off Inform instruction:

```xml
<INFORM>
    <GCODE func="G90">
    <Header>False</Header>
    <Description>Linear Motion</Description>
    <isLiteralInform>True</isLiteralInform>
    <MotionFunction>False</MotionFunction>
    </GCODE>
</INFORM>
```

<table>
<thead>
<tr>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>complexType INFORM</td>
</tr>
</tbody>
</table>
GenInform is an abbreviation for General Inform. Any valid Inform command can be used inside the GenInform Command but it is the responsibility of the user to know the correct syntax. This is a pass through command that is not syntax checked.

This command allows users to use any map an G Code Code command into any Inform Instruction. It is used in the GTOINFORM-- INFORM section of the Configuration File to map a G Code Instruction into a Robot Inform Instruction.

Example:
The following example maps a G90 Instruction into a Shift Off Inform instruction;

```xml
<GTOINFORM>
  <GCODE func="G90">
    <Header>False</Header>
    <Description>Linear Motion</Description>
    <isLiteralInform>True</isLiteralInform>
    <MotionFunction>False</MotionFunction>
  </GCODE>
  <INFORM>
    <GenInform>SFTOF</GenInform>
  </INFORM>
</GTOINFORM>
```

Source

```xml
<xs:element name="GenInform" type="xs:string">
  <xs:annotation>
    <xs:documentation>
      GenInform is an abbreviation for General Inform. Any valid Inform command can be used inside the GenInform Command but it is the responsibility of the user to know the correct syntax. This is a pass through command that is not syntax checked.

      This command allows users to use any map an G Code Code command into any Inform Instruction. It is used in the GTOINFORM-- INFORM section of the Configuration File to map a G Code Instruction into a Robot Inform Instruction.

      Example:
The following example maps a G90 Instruction into a Shift Off Inform instruction;

      ```xml
        <GTOINFORM>
          <GCODE func="G90">
            <Header>False</Header>
            <Description>Linear Motion</Description>
            <isLiteralInform>True</isLiteralInform>
            <MotionFunction>False</MotionFunction>
          </GCODE>
          <INFORM>
            <GenInform>SFTOF</GenInform>
          </INFORM>
        </GTOINFORM>
      ```
    </xs:documentation>
  </xs:annotation>
</xs:element>
```
Each GTOINFORM Instruction has two sub elements called GCODE and INFORM. The G Code part of the instruction identifies the G Code instruction that will be converted into Inform. The Inform Instruction gives the XML Commands that are used to create the Inform Instructions.

NOTE: If a G Code Instruction is not listed in one of the GCODE elements, that instruction will be ignored.

```
<xsl:element name="GTOINFORM" type="GCodeToInform">
  <xsl:annotation>
    <xsl:documentation>
      Each GTOINFORM Instruction has two sub elements called GCODE and INFORM. The G Code part of the instruction identifies the G Code instruction that will be converted into Inform. The Inform Instruction gives the XML Commands that are used to create the Inform Instructions.
      
      NOTE: If a G Code Instruction is not listed in one of the GCODE elements, that instruction will be ignored.
    </xsl:documentation>
  </xsl:annotation>
</xsl:element>
```
<table>
<thead>
<tr>
<th>type</th>
<th>restriction of <code>xs:string</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td></td>
<td>default False</td>
</tr>
<tr>
<td>used by</td>
<td>complexType <strong>GCODE</strong></td>
</tr>
<tr>
<td>facets</td>
<td>enumeration False</td>
</tr>
<tr>
<td></td>
<td>enumeration True</td>
</tr>
<tr>
<td>annotation</td>
<td>documentation</td>
</tr>
<tr>
<td></td>
<td>This instruction is used in the G Code Tag. It indicates if the G Code function will map into an Inform Header Instruction. Values may be True or False.</td>
</tr>
<tr>
<td>source</td>
<td><code>&lt;xs:element name=&quot;Header&quot; default=&quot;False&quot;&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:annotation&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:documentation&gt;</code></td>
</tr>
<tr>
<td></td>
<td>This instruction is used in the G Code Tag. It indicates if the G Code function will map into an Inform Header Instruction. Values may be True or False.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;/xs:documentation&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:simpleType&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:restriction base=&quot;xs:string&quot;&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;False&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;True&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/xs:restriction&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/xs:simpleType&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/xs:element&gt;</code></td>
</tr>
</tbody>
</table>

**element** **INFORM**

---

**Diagram**

**MotionType**

MotionType allows the user to specify the type of robotic motion that will be used to send the robot to a position. The MotionType can be Circular, Joint, or Linear. Circular Motion will move the robot in a circular arc. Joint motion will move all robot axes and Linear will move the robot in a straight line.

**OutputOn**

This command is used to map a G or M Code into a robot Output On Command.

**Timer**

Create a time delay.
Timer

Creates a time delay Inform Function. Time delay is in seconds.

ThreeDOF

Supplies the default orientation of the Robot Tool. This is the value that makes the last length of tool perpendicular to the part work plane.

LengthUnits

ReferenceModel

OutputOff

0..∞

Turns Off a robot Output whose number is specified in the OutputOff Tag.

GenInform

GenInform is an abbreviation for General Inform. Any valid Inform command can be used inside the GenInform Command but it is the responsibility of the user to know the correct syntax. This is a pass through command that is not syntax checked.

This command allows users to use any map an G Code Code command into any Inform Instruction. It is used in the GTOInform-- INFORM section of the Configuration File to map a G Code Instruction into a Robot Inform Instruction.

Example:
The following example maps a G90 Instruction into a Shift Off Inform instruction:

<GTOInform>
  <GCODE
    func="G90">
    <Header>False</Header>
    <Description>Linear Motion</Description>
    <isLiteralInform>True</isLiteralInform>
    <MotionFunction>False</MotionFunction>
  </GCODE>
  <INFORM>
<GCODE>
<INFORM>
<br>CallJob<br>
This is an XML to Inform Instruction. Use this element to cause the robot to call a job subroutine.
<br>This instruction will appear in the intermediate XML file that the G Code Converter creates during the conversion process. It will produce an Inform Instruction in the robot JBI with<br>Example use:<br><CallJob>Pickup</CallJob><br>Example output JBI: CALL JOB:Pickup<br>1..∞
</INFORM>
</GCODE>

INFORM
This is a sub element of the GTOINFORM Element. It defines the associated G Code Instruction will be converted into Inform.

ShiftOffset
Maps a G Code to a shift On or Shift Setup Function.

Velocity
This element is used in the Position element. It gives the velocity which the robot will use to move from its current position to the specified position.
Note: If the Motion Type is Linear, the Velocity has a range of about 0 to 1600 millimeters/sec. If the Motion type is Joint the Velocity has a range from 0 to 100%.

PosLevel
Position Level tells the controller how close to come to the programmed location. This is used to allow the robot to round or smooth direction changes. Level 0 makes the robot most accurate. Level 5 makes the smoothest path.

MotionFunction
This command is used to list all of the G Code functions that will cause robot motion. It is used by the G Code Converter during the G Code to Inform Conversion to parse all commands with robot motion.
Code to Inform Conversion to parse all commands with multiple instructions and place the motion functions last, next to the XYZ position values.

**ShiftSourceNum**

P Variable Index Number

**SetVar**

SetVar is an XML to JBI Instruction that allows the setting of Robot Variables. These Variables may be B, I, D or R variables.

P variables are set with SetPosElmnt.

This Element may have one of two forms. In form one, you specify the Variable Type and Index and provide the variable value with a "Const" element. In Form 2 you specify the source and destination variable types and indices.

**SetPosElmnt**

**ShiftOn**

Tells the robot the reference frame of the position. This can be:
RF for Robot Frame,
BF for Base Frame,
TF for tool Frame,
UFx for User Frame, or it can point to a model in the MotoSmEG Cell.

**InformMotionFunctionGroup**

This command allows the user to create more than one Motion Function for a single G Code Command.

Alternatively the user can call a job and the job can have multiple motion positions.

Example:
```
<InformMotionFunctionGroup
MotionType="Linear" ZValue="Z"
PosLevel="0"
CannedVelocityRegister="1"/>
<InformMotionFunctionGroup
MotionType="Inc" ZValue="Z"
PosLevel="0" PosVar="121"
FillWithDVar="1" ExtraOffsetDVar="2"
RefFrame="Tool"/>
<InformMotionFunctionGroup
MotionType="Inc" ZValue="R"
PosLevel="0"
CannedVelocityRegister="1"
PosVar="122" FillWithDVar="1"
ExtraOffsetDVar="2"/>
```
<table>
<thead>
<tr>
<th>type</th>
<th>INFORM</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content complex</td>
</tr>
<tr>
<td>children</td>
<td>MotionType OutputOn Timer ThreeDOF LengthUnits ReferenceModel OutputOff GenInform CallJob ShiftOffset Velocity PosLevel MotionFunction ShiftSourceNum SetVar SetPosElmnt ShiftOn InformMotionFunctionGroup RotatingStationPosition Position</td>
</tr>
<tr>
<td>used by</td>
<td>complexType GCodeToInform</td>
</tr>
<tr>
<td>annotation documentation</td>
<td>This is a sub element of the GTOINFORM Element. It defines the associated G Code Instruction will be converted into Inform.</td>
</tr>
</tbody>
</table>

```xml
<xs:element name="INFORM" type="INFORM">
  <xs:documentation>This is a sub element of the GTOINFORM Element. It defines the associated G Code Instruction will be converted into Inform.<br/>
</xs:documentation>
</xs:element>
```

element InformMotionFunctionGroup

diagram
type extension of InformMotionFunctionGroupType

properties content complex

used by complexTypes FileConfigurationType INFORM

<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MotionType</td>
<td>derived by: xs:string</td>
<td>optional</td>
<td>Linear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZValue</td>
<td>derived by: xs:string</td>
<td>optional</td>
<td>Z</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PosLevel</td>
<td>derived by: xs:integer</td>
<td>optional</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CannedVelocityRegister</td>
<td>derived by: xs:integer</td>
<td>optional</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PosVar</td>
<td>derived by: xs:integer</td>
<td>optional</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FillWithDVar</td>
<td>derived by: xs:integer</td>
<td>optional</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ExtraOffsetDVar</td>
<td>derived by: xs:integer</td>
<td>optional</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RefFrame</td>
<td>xs:anySimpleType</td>
<td>optional</td>
<td>Tool</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This command allows the user to create more than one Motion Function for a single G Code Command.

Alternatively the user can call a job and the job can have multiple motion positions.

Example:
<InformMotionFunctionGroup MotionType="Linear" ZValue="Z" PosLevel="0" CannedVelocityRegister="1"/>
<InformMotionFunctionGroup MotionType="Inc" ZValue="Z" PosLevel="0" PosVar="121" FillWithDVar="1" ExtraOffsetDVar="2" RefFrame="Tool"/>
<InformMotionFunctionGroup MotionType="Inc" ZValue="R" PosLevel="0" CannedVelocityRegister="1" PosVar="122" FillWithDVar="1" ExtraOffsetDVar="2" RefFrame="Tool"/>

attribute InformMotionFunctionGroup/@RefFrame

type xs:anySimpleType

properties
isRef 0
default Tool
use optional

source <xs:attribute name="RefFrame" type="xs:anySimpleType" use="optional" default="Tool"/>

element InputPositionData

source <xs:complexType>
  <xs:simpleContent>
    <xs:extension base="InputPositionDataType">
      <xs:attribute name="InputPositionDataPulse" type="xs:boolean" use="optional"/>
      <xs:attribute name="InputPositionDataTypeRectan" type="xs:boolean" use="optional"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
element `isLiteralInform`

```xml
<xs:element name="isLiteralInform" default="False">
    <xs:annotation>
        <xs:documentation>When true, this command allows the XML commands in the corresponding Inform tag to be placed in the XML intermediate file without any further processing.</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="False"/>
            <xs:enumeration value="True"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

### element `JobOutputOptions`

```xml
<xs:complexType name="JobOutputOptionsType">
    <xs:sequence>
        <xs:element name="JobType"/>
    </xs:sequence>
</xs:complexType>
```
element L

<table>
<thead>
<tr>
<th>diagram</th>
<th>📜 L</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>xs:integer</td>
</tr>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>complexType ArmConfigType</td>
</tr>
</tbody>
</table>

source `<xs:element name="L" type="xs:integer"/>

element LengthUnits

<table>
<thead>
<tr>
<th>diagram</th>
<th>📜 LengthUnits</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:string</td>
</tr>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>complexTypes INFORM UnitsType</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration Inches</td>
</tr>
<tr>
<td></td>
<td>enumeration Millimeters</td>
</tr>
</tbody>
</table>

source `<xs:element name="LengthUnits">
  `<xs:simpleType>
    `<xs:restriction base="xs:string">
      `<xs:enumeration value="Inches"/>
      `<xs:enumeration value="Millimeters"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

element MotionFunction

<table>
<thead>
<tr>
<th>diagram</th>
<th>📜 MotionFunction</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:string</td>
</tr>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>complexTypes GCODE INFORM MotionFunctionsType</td>
</tr>
</tbody>
</table>
This command is used to list all of the G Code functions that will cause robot motion. It is used by the G Code Converter during the G Code to Inform Conversion to parse all commands with multiple instructions and place the motion functions last, next to the XYZ position values.

```
<xs:element name="MotionFunction">
  <xs:annotation>
    <xs:documentation>This command is used to list all of the G Code functions that will cause robot motion. It is used by the G Code Converter during the G Code to Inform Conversion to parse all commands with multiple instructions and place the motion functions last, next to the XYZ position values.</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="5"/>
      <xs:minLength value="1"/>
      <xs:enumeration value="False"/>
      <xs:enumeration value="G0"/>
      <xs:enumeration value="G00"/>
      <xs:enumeration value="G01"/>
      <xs:enumeration value="G02"/>
      <xs:enumeration value="G03"/>
      <xs:enumeration value="G1"/>
      <xs:enumeration value="G2"/>
      <xs:enumeration value="G3"/>
      <xs:enumeration value="G81"/>
      <xs:enumeration value="True"/>
      <xs:enumeration value="False"/>
      <xs:enumeration value="Linear"/>
      <xs:enumeration value="Inc"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```
<table>
<thead>
<tr>
<th>properties</th>
<th>content simple</th>
</tr>
</thead>
<tbody>
<tr>
<td>facets</td>
<td>enumeration False</td>
</tr>
<tr>
<td></td>
<td>enumeration G0</td>
</tr>
<tr>
<td></td>
<td>enumeration G00</td>
</tr>
<tr>
<td></td>
<td>enumeration G01</td>
</tr>
<tr>
<td></td>
<td>enumeration G02</td>
</tr>
<tr>
<td></td>
<td>enumeration G03</td>
</tr>
<tr>
<td></td>
<td>enumeration G1</td>
</tr>
<tr>
<td></td>
<td>enumeration G2</td>
</tr>
<tr>
<td></td>
<td>enumeration G3</td>
</tr>
<tr>
<td></td>
<td>enumeration G81</td>
</tr>
<tr>
<td></td>
<td>enumeration True</td>
</tr>
<tr>
<td></td>
<td>enumeration False</td>
</tr>
</tbody>
</table>

```
source <xs:element name="MotionFunctionList">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="False"/>
      <xs:enumeration value="G0"/>
      <xs:enumeration value="G00"/>
      <xs:enumeration value="G01"/>
      <xs:enumeration value="G02"/>
      <xs:enumeration value="G03"/>
      <xs:enumeration value="G1"/>
      <xs:enumeration value="G2"/>
      <xs:enumeration value="G3"/>
      <xs:enumeration value="G81"/>
      <xs:enumeration value="True"/>
      <xs:enumeration value="False"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

element **MotionGroup**

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="MotionGroup" /></td>
</tr>
<tr>
<td>type</td>
</tr>
<tr>
<td>properties</td>
</tr>
<tr>
<td>used by</td>
</tr>
<tr>
<td>facets</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>annotation</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
**element MotionType**

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of <code>xs:string</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>element <strong>PositionType</strong></td>
</tr>
<tr>
<td></td>
<td>complexType <strong>INFORM</strong></td>
</tr>
<tr>
<td>facets</td>
<td>enumeration Circular</td>
</tr>
<tr>
<td></td>
<td>enumeration Joint</td>
</tr>
<tr>
<td></td>
<td>enumeration Linear</td>
</tr>
<tr>
<td></td>
<td>enumeration SynchLinear</td>
</tr>
<tr>
<td></td>
<td>enumeration SynchCircular</td>
</tr>
<tr>
<td>annotation</td>
<td>documentation <strong>MotionType</strong> allow the user to specify the type of robotic motion that will be used to send the robot to a position. The MotionType can be Circular, Joint, or Linear. Circular Motion will move the robot in a circular arc. Joint motion will time coordinate all robot axes and Linear will move the robot in a straight line.</td>
</tr>
</tbody>
</table>
<xs:element name="MotionType">
   <xs:annotation>
      <xs:documentation>MotionType allow the user to specify the type of robotic motion that will be used to
send the robot to a position. The MotionType can be Circular, Joint, or Linear.
Circular Motion will move the robot in a circular arc. Joint motion will time coordinate all robot axes and
Linear will move the robot in a straight line.</xs:documentation>
   </xs:annotation>
   <xs:simpleType>
      <xs:restriction base="xs:string">
         <xs:enumeration value="Circular"/>
         <xs:enumeration value="Joint"/>
         <xs:enumeration value="Linear"/>
         <xs:enumeration value="SynchLinear"/>
         <xs:enumeration value="SynchCircular"/>
      </xs:restriction>
   </xs:simpleType>
</xs:element>

**element MoveTag**

- **diagram**
- **type** restriction of `xs:string`
- **properties** content simple
  - default NWAIT
- **used by** element `PositionType`
- **facets** enumeration NWAIT

```
<xs:element name="MoveTag" default="NWAIT">
   <xs:simpleType>
      <xs:restriction base="xs:string">
         <xs:enumeration value="NWAIT"/>
      </xs:restriction>
   </xs:simpleType>
</xs:element>
```

**element OutputOff**

- **diagram**
- **type** `xs:byte`
- **properties** content simple
- **used by** complexType `INFORM`
- **annotation** documentation
  - Turns Off a robot Output whose number is specified in the OutputOff Tag.

```
<xs:element name="OutputOff">
   <xs:simpleType>
      <xs:restriction base="xs:byte"/>
   </xs:simpleType>
</xs:element>
```
source <xs:element name="OutputOff" type="xs:byte">
  <xs:annotation>
    <xs:documentation>Turns Off a robot Output whose number is specified in the OutputOff Tag.</xs:documentation>
  </xs:annotation>
</xs:element>

element OutputOn

diagram

This command is used to map a G or M Code into a robot Output On Command.

type xs:byte

properties content simple
default 1

used by complexType INFORM

annotation documentation
This command is used to map a G or M Code into a robot Output On Command.

source <xs:element name="OutputOn" type="xs:byte" default="1">
  <xs:annotation>
    <xs:documentation>This command is used to map a G or M Code into a robot Output On Command.</xs:documentation>
  </xs:annotation>
</xs:element>

element PositionType

diagram

MotionType allow the user to specify the type of robotic motion that will be used to send the robot to a position. The MotionType can be Circular, Joint, or Linear. Circular Motion will move the robot in a circular arc. Joint motion will time coordinate all robot axes and Linear will move the robot in a straight line.

Cartesian coordinates can include XYZ or XYZRxRyRz. If Rx, Ry, and Rz are not specified, the default ThreeDOF Data will be used.
PositionType

This is the element that specifies the move positions, Velocities, motion types and other ancillary position information.

Velocity

PosLevel

Position Level tells the controller how close to come to the programmed location. This is used to allow the robot to round or smooth direction changes. Level 0 makes the robot most accurate. Level 5 makes the smoothest path.

ReFrame

Tells the robot the reference frame of the position. This can be:
- RF for Robot Frame.
- BF for Base Frame.
- TF for tool Frame.
- UFx for User Frame, or it can point to a model in the MotoSmEG Cell.

Tool

Specifies the Tool Number to user for this move.

Form

XXXFillIn

Search

This command is used to allow the robot to move until a variable is On or Off. Maximum Search distance is specified in the Const Tag. The Actual Search Distance in placed in the D Variable whose index is specified in the DistVar tag.

NOTE: Only works with IMOV commands.
<table>
<thead>
<tr>
<th>properties</th>
<th>content</th>
<th>complex</th>
</tr>
</thead>
<tbody>
<tr>
<td>children</td>
<td></td>
<td>MotionType X Y Z Ry Rx S L U B T PosVar Velocity PosLevel RefFrame Tool Form Search MoveTag RotatingStationPosition</td>
</tr>
<tr>
<td>annotation</td>
<td>documentation</td>
<td>This is the element that specifies the move positions, Velocities, motion types and other ancillary position information.</td>
</tr>
</tbody>
</table>

```xml
<xs:element name="PositionType">
    <xs:annotation>
        <xs:documentation>This is the element that specifies the move positions, Velocities, motion types and other ancillary position information.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element ref="MotionType" minOccurs="0"/>
        </xs:choice>
        <xs:sequence minOccurs="0">
            <xs:annotation>
                <xs:documentation>Robot positions can be in Cartesian Coordinates, Joint Coordinates or in a P Variable. P variable maybe in either Joint or Cartesian.</xs:documentation>
            </xs:annotation>
            <xs:choice minOccurs="0"/>
        </xs:sequence>
        <xs:sequence minOccurs="0">
            <xs:element name="X" minOccurs="0"/>
            <xs:element name="Y" minOccurs="0"/>
            <xs:element name="Z" minOccurs="0"/>
            <xs:element name="Ry" minOccurs="0"/>
            <xs:element name="Rx" minOccurs="0"/>
            <xs:element name="Rz" minOccurs="0"/>
        </xs:sequence>
        <xs:sequence minOccurs="0">
            <xs:element name="S" type="xs:short"/>
            <xs:element name="L" type="xs:short"/>
            <xs:element name="U" type="xs:short"/>
            <xs:element name="B" type="xs:short"/>
            <xs:element name="T" type="xs:short"/>
        </xs:sequence>
        <xs:sequence minOccurs="0">
            <xs:element name="PosVar" default="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```
<xs:restriction base="xs:integer">
    <xs:totalDigits value="3"/>
</xs:restriction>
</xs:element>

<xs:element name="Velocity" minOccurs="0">
    <xs:restriction base="xs:decimal">
        <xs:totalDigits value="8"/>
        <xs:fractionDigits value="3"/>
    </xs:restriction>
</xs:element>

<xs:element ref="PosLevel" minOccurs="0"/>
<xs:element ref="RefFrame" minOccurs="0"/>

<xs:element name="Tool" type="xs:integer" default="0" minOccurs="0">
    <xs:annotation>
        <xs:documentation>Specifies the Tool Number to user for this move.</xs:documentation>
    </xs:annotation>
</xs:element>

<xs:element name="Form" type="xs:integer" minOccurs="0">
    <xs:annotation>
        <xs:documentation>XXXFillIn</xs:documentation>
    </xs:annotation>
</xs:element>

<xs:element name="Search" type="Search" minOccurs="0">
    <xs:annotation>
        <xs:documentation>This command is used to allow the robot to move until a variable is On or Off. Maximum Search distance is specified in the Const Tag. The Actual Search Distance in placed in the D Variable whose index is specified in the DistVar tag.

        NOTE: Only works with IMOV commands.

        Inform Syntax is:
        SRCH RIN#(x) = y DIS=Z Dw where:
        x = Input
        y=State
        z=Const
        w=DistVar
    </xs:documentation>
</xs:annotation>
</xs:element>

<xs:element ref="MoveTag" minOccurs="0"/>
<xs:element name="RotatingStationPosition" type="RotatingStationJobDataPositionType" minOccurs="0" maxOccurs="4"/>
</xs:choice>
</xs:complexType>
</xs:element>
### element PositionType/Y

```xml
<xs:element name="Y" minOccurs="0"/>
```

### element PositionType/Z

```xml
<xs:element name="Z" minOccurs="0"/>
```

### element PositionType/Ry

```xml
<xs:element name="Ry" minOccurs="0"/>
```

### element PositionType/Rx

```xml
<xs:element name="Rx" minOccurs="0"/>
```

### element PositionType/Rz
<table>
<thead>
<tr>
<th>properties</th>
<th>isRef 0 \nminOcc 0 \nmaxOcc 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td><code>&lt;xs:element name=&quot;Rz&quot; minOccurs=&quot;0&quot;/&gt;</code></td>
</tr>
</tbody>
</table>

**element PositionType/S**

```
+---+
| S |
+---+
```

- **type**: `xs:short`
- **properties**: `isRef 0` \n  `content simple`
- **used by**: `complexType ArmConfigType`
- **source**: `<xs:element name="S" type="xs:short"/>`

**element PositionType/L**

```
+---+
| L |
+---+
```

- **type**: `xs:short`
- **properties**: `isRef 0` \n  `content simple`
- **used by**: `complexType ArmConfigType`
- **source**: `<xs:element name="L" type="xs:short"/>`

**element PositionType/U**

```
+---+
| U |
+---+
```

- **type**: `xs:short`
- **properties**: `isRef 0` \n  `content simple`
- **used by**: `complexType ArmConfigType`
- **source**: `<xs:element name="U" type="xs:short"/>`

**element PositionType/B**

```
+---+
| B |
+---+
```

- **type**: `xs:short`
- **properties**: `isRef 0` \n  `content simple`
- **used by**: `complexType ArmConfigType`
**element PositionType/T**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="Diagram" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td><code>xs:short</code></td>
</tr>
</tbody>
</table>
| properties | `isRef 0`  
|          | `content simple` |
| used by | complexType `ArmConfigType` |
| source  | `<xs:element name="T" type="xs:short"/>` |

**element PositionType/PosVar**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="Diagram" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of <code>xs:integer</code></td>
</tr>
</tbody>
</table>
| properties | `isRef 0`  
|          | `content simple`  
|          | `default 0`  |
| facets  | `totalDigits 3`  
|          | `fractionDigits 0`  |
| source  | `<xs:element name="PosVar" default="0">  
|          | `<xs:simpleType>`  
|          | `<xs:restriction base="xs:integer">  
|          | `<xs:fractionDigits value="0"/>`  
|          | `<xs:totalDigits value="3"/>`  
|          | `</xs:restriction>`  
|          | `</xs:simpleType>`  
|          | `</xs:element>`  |

**element PositionType/Velocity**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="Diagram" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of <code>xs:decimal</code></td>
</tr>
</tbody>
</table>
| properties | `isRef 0`  
|          | `minOcc 0`  
|          | `maxOcc 1`  
|          | `content simple` |
| used by | complexType `INFORM` |
| facets  | `totalDigits 8`  
|          | `fractionDigits 3`  |
**element PositionType/Tool**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="Diagram" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td><code>xs:integer</code></td>
</tr>
</tbody>
</table>
| properties | isRef 0  
|           | minOcc 0  
|           | maxOcc 1  
| content  | simple    |
| default  | 0          |
| annotation documentation | Specifies the Tool Number to user for this move. |

**source**

```
<xs:element name="Tool" type="xs:integer" default="0" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Specifies the Tool Number to user for this move.</xs:documentation>
  </xs:annotation>
</xs:element>
```

---

**element PositionType/Form**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="Diagram" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td><code>xs:integer</code></td>
</tr>
</tbody>
</table>
| properties | isRef 0  
|           | minOcc 0  
|           | maxOcc 1  
| content  | simple    |
| annotation documentation | `XXXFillIn` |

**source**

```
<xs:element name="Form" type="xs:integer" minOccurs="0">
  <xs:annotation>
    <xs:documentation>XXXFillIn</xs:documentation>
  </xs:annotation>
</xs:element>
```

---

**element PositionType/Search**

---
This command is used to allow the robot to move until a variable is On or Off. Maximum Search distance is specified in the Const Tag. The Actual Search Distance in placed in the D Variable whose index is specified in the DistVar tag.

NOTE: Only works with IMOV commands.

Inform Syntax is:
SRCH RIN#(x) = y DIS=Z Dw where:
x = Input
y=State
z=Const
w=DistVar
This command is used to allow the robot to move until a variable is On or Off. Maximum Search distance is specified in the Const Tag. The Actual Search Distance in placed in the D Variable whose index is specified in the DistVar tag.

NOTE: Only works with IMOV commands.

Inform Syntax is:
SRCH RIN#(x) = y DIS=Z Dw where:
x = Input
y=State
z=Const
w=DistVar

---

**element** PositionType/RotatingStationPosition

**source**

```xml
<xs:element name="Search" type="Search" minOccurs="0">
  <xs:annotation>
    <xs:documentation>This command is used to allow the robot to move until a variable is On or Off. Maximum Search distance is specified in the Const Tag. The Actual Search Distance in placed in the D Variable whose index is specified in the DistVar tag.

    NOTE: Only works with IMOV commands.

    Inform Syntax is:
    SRCH RIN#(x) = y DIS=Z Dw where:
    x = Input
    y=State
    z=Const
    w=DistVar</xs:documentation>
    </xs:annotation>
  </xs:element>
```

**element** PosLevel
**element **PosLevel**

| diagram | Position Level tells the controller how close to come to the programmed location. This is used to allow the robot to round or smooth direction changes. Level 0 makes the robot most accurate. Level 5 makes the smoothest path. |

| type | xs:integer |
| properties | content simple default 3 |
| used by | element PositionType complexType INFORM |
| annotation | documentation Position Level tells the controller how close to come to the programmed location. This is used to allow the robot to round or smooth direction changes. Level 0 makes the robot most accurate. Level 5 makes the smoothest path. |
| source | `<xs:element name="PosLevel" type="xs:integer" default="3">  
  <xs:annotation>  
    <xs:documentation>Position Level tells the controller how close to come to the programmed location. This is used to allow the robot to round or smooth direction changes. Level 0 makes the robot most accurate. Level 5 makes the smoothest path. </xs:documentation>  
  </xs:annotation>  
</xs:element>` |

**element R**

| diagram |  |
| type | xs:integer |
| properties | content simple |
| used by complexType ArmConfigType |
| source | `<xs:element name="R" type="xs:integer"/>` |

**element ReferenceModel**

<p>| diagram |  |
| properties | content complex mixed true |
| used by complexType INFORM |</p>
<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JobTypeMaster</td>
<td>derived by: xs:string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>JobTypeSub</td>
<td>derived by: xs:string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```xml
<source>
<xs:element name="ReferenceModel">
  <xs:complexType mixed="true">
    <xs:attribute name="JobTypeMaster">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="JobTypeSub">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>

attribute ReferenceModel/@JobTypeMaster

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration True</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:attribute name=&quot;JobTypeMaster&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:simpleType</a></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:restriction base=&quot;xs:string&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;True&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:restriction&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:simpleType&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:attribute&gt;</td>
</tr>
</tbody>
</table>

attribute ReferenceModel/@JobTypeSub

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration True</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:attribute name=&quot;JobTypeSub&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:simpleType</a></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:restriction base=&quot;xs:string&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;True&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:restriction&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:simpleType&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:attribute&gt;</td>
</tr>
</tbody>
</table>
## element RefFrame

<table>
<thead>
<tr>
<th>Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="RefFrame" /></td>
</tr>
</tbody>
</table>

Tells the robot the reference frame of the position. This can be:
- RF for Robot Frame,
- BF for Base Frame,
- TF for tool Frame,
- UFx for User Frame, or it can point to a model in the MotoSmEG Cell.

### Type

restriction of xs:string

### Properties

content simple

### Used by

elements **PositionType** **ShiftOn**

### Facets

- enumeration BF
- enumeration RF
- enumeration TF
- enumeration UF1
- enumeration UF2
- enumeration UF3
- enumeration UF4
- enumeration UF5
- enumeration UF6
- enumeration UF7
- enumeration UF8
- enumeration UF9
- enumeration UF10
- enumeration UF11
- enumeration UF12
- enumeration UF13
- enumeration UF14
- enumeration UF15
- enumeration UF16
- enumeration UF17
- enumeration UF18
- enumeration UF19
- enumeration UF20
- enumeration UF21
- enumeration UF22
- enumeration UF23
- enumeration UF24
- enumeration MTF

### Annotation

documentation

Tells the robot the reference frame of the position. This can be:
- RF for Robot Frame,
- BF for Base Frame,
- TF for tool Frame,
- UFx for User Frame, or it can point to a model in the MotoSmEG Cell.
Tells the robot the reference frame of the position. This can be:
RF for Robot Frame,
BF for Base Frame,
TF for tool Frame,
UFx for User Frame, or it can point to a model in the MotoSmEG Cell.
This is a required element.

This is a Regular Expression instruction. This is the instruction tells the G Code Converter which G Code Instructions to look for and the format of the G Code file.

This instruction is written in a way that most, if not all, G Code files can be converted without modifying this instruction. Many G Code fields are optional and may or may not appear in the G Code file. For example line numbers may or may not be included in the G Code instruction.

Below is a RegExParseInt from one of the examples.

```
```

It can be broken down into groups of instructions. The line number parsing instruction is shown below:

```
(N[0-9]+)\s*
```

This instruction says to look for a N followed by 1 to 5 numerical digits. The + indicates that the value must occur 1 or more times to match and the \s says that the NXXXX may be followed by a white space character, a tab, form-feed or carriage return.

The M or G command is parsed by the follow RegEx expression:

```
([Mm][Gg][0-9]+)\s*
```

This expression say to look for an M or a G followed by 1 to 3 numerical digits. The * says that match may occur 0 or more times. The \s* indicates that white spaces may or may not be included after the M or G instruction.
<table>
<thead>
<tr>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>complexType FileConfigurationType</td>
</tr>
<tr>
<td>annotation</td>
<td>documentation</td>
</tr>
</tbody>
</table>

This is a required element.

This is a Regular Expression instruction. This is the instruction tells the G Code Converter which G Code Instructions to look for and the format of the G Code file.

This instruction is written in a way that most, if not all, G Code files can be converted without modifying this instruction. Many G Code fields are optional and may or may not appear in the G Code file. For example line Numbers may or may not be included in the G Code instructions.

Below is a RegExParseInt from one of the examples.

```
<RegExParseInt>(N\d\{1,5\})\+\s(MG)\d\{1,3\}\s*X\d\{0,3\}\s*Y\d\{0,4\}\s*Z\d\{0,4\}\s*I\d\{0,3\}\s*J\d\{0,4\}\s*A\d\{0,4\}\s*B\d\{0,4\}\s*F\d\{0,4\}\s*</RegExParseInt>
```

It can be broken down into groups of instructions. The line number parsing instruction is shown below:

```
(N\d\{1,5\})\+\s
```

This instruction says to look for a N followed by 1 to 5 numerical digits. The + indicates that the value must occur 1 or more times to match and the \s says that the NXXXX may be followed by a white space character, a tab, form-feed or carriage return.

The M or G command is parsed by the following RegEx expression:

```
(MG)\d\{1,3\}\s*
```

This expression says to look for an M or a G followed by 1 to 3 numerical digits. The * says that match may occur 0 or more times. The \s* indicates that white spaces may or may not be included after the M or G instruction.

The X location instruction is parsed with the following command:

```
*([X]\d\{0,3\}\s*\d\{0,4\}\s*)
```
This instruction is similar to the ones above except that the X may or may not include a – or negative sign, 0 to 3 digits, a . or period may or may not be included followed by 0 to 4 additional digits.

More information is available on the Web at http://regex.osherove.com

NOTE: Corruption of this line may disable the G Code Converter. Please make a backup of the original instruction before you make any modifications to it.

<xs:element name="RegExParseInt" type="xs:string">
  <xs:annotation>
    <xs:documentation>This is a required element.

    This is a Regular Expression instruction. This is the instruction tells the G Code Converter which G Code Instructions to look for and the format of the G Code file.

    This instruction is written in a way that most, if not all, G Code files can be converted without modifying this instruction. Many G Code fields are optional and may or may not appear in the G Code file. For example line Numbers may or may not be included in the G Code instruction.

    Below is a RegExParseInt from one of the examples.

    <RegExParseInt>(N\d{1,5})+\s([MG]\d{1,3})\s*([D][-]*\d{0,3})\s*([X][-]*\d{0,3}\.\d{0,4})\s*([Y][-]*\d{0,3}\.\d{0,4})\s*([I][-]*\d{0,3}\.\d{0,4})\s*([J][-]*\d{0,3}\.\d{0,4})\s*([A][-]*\d{0,3}\.\d{0,4})\s*([B][-]*\d{0,3}\.\d{0,4})\s*([F][-]*\d{0,3}\.\d{0,4})*</RegExParseInt>

    It can be broken down into groups of instructions. The line number parsing instruction is shown below:

    (N\d{1,5})+\s

    This instruction says to look for a N followed by 1 to 5 numerical digits. The + indicates that the value must occur 1 or more times to match and the \s says that the NXXXX may be followed by a white space character, a tab, form-feed or carriage return.

    The M or G command is parsed by the follow RegEx expression:

    ([MG]\d{1,3})\s*

    This expression say to look for an M or a G followed by 1 to 3 numerical digits. The * says that match may occur 0 or more times. The \s* indicates that white spaces may or may not be included after the M or G instruction.

    The X location instruction is parsed with the following command;

    *([X][-]*\d{0,3}\.\d{0,4})*\s*

    This instruction is similar to the ones above except that the X may or may not include a – or negative sign, 0 to 3 digits, a . or period may or may not be included followed by 0 to 4 additional digits.

    More information is available on the Web at http://regex.osherove.com

    NOTE: Corruption of this line may disable the G Code Converter. Please make a backup of the original instruction before you make any modifications to it.

  </xs:documentation>
</xs:element>
### element Root

<table>
<thead>
<tr>
<th>properties</th>
<th>content</th>
<th>complex</th>
</tr>
</thead>
</table>

| children | FileConfiguration GTOINFORM |

<table>
<thead>
<tr>
<th>annotation</th>
<th>documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>This is a required element. It is the base or start of the XML Configuration data.</td>
<td></td>
</tr>
</tbody>
</table>

| source | `<xs:element name="Root">`  
|        | `<xs:annotation>`  
|        | `<xs:documentation>This is a required element. It is the base or start of the XML Configuration data.</xs:documentation>`  
|        | `<xs:annotation>`  
|        | `<xs:complexType>`  
|        | `<xs:sequence>`  
|        | `<xs:element name="FileConfiguration" type="FileConfigurationType"/>`  
|        | `<xs:element ref="GTOINFORM" maxOccurs="unbounded"/>`  
|        | `<xs:sequence>`  
|        | `<xs:complexType>`  
|        | `<xs:element>` |

### element Root/FileConfiguration
Figure Configuration Type

RegExParseInt

This is a required element.

This is a Regular Expression instruction. This is the instruction tells the G Code Converter which G Code Instructions to look for and the format of the G Code file.

This instruction is written in a way that most, if not all, G Code files can be converted without modifying this instruction. Many G Code fields are optional and may or may not appear in the G Code file. For example line Numbers may or may not be included in the G Code instruction.

Below is a RegExParseInt from one of the examples.

```
<RegExParseInt>(\d{1,5})
+\s*(\[(\d{1,3})*\])\s*(\[(D|L|T)])
\s*(\d{0,3})\s*(\d{0,4})\s*(\[(X|Y|Z|F)])
\s*(\d{0,3})\s*(\d{0,4})\s*(\[(H|T)])
\s*(\d{0,3})\s*(\d{0,4})\s*(\[(I)])
\s*(\d{0,3})\s*(\d{0,4})\s*(\[(J)])
\s*(\d{0,3})\s*(\d{0,4})\s*(\[(A|C)])
\s*(\d{0,3})\s*(\d{0,4})\s*(\[(B)])
\s*(\d{0,3})\s*(\d{0,4})\s*(\[(F)])
\s*(\d{0,3})\s*(\d{0,4})\s*(\[])</RegExParseInt>
```

It can be broken down into groups of instructions. The line number parsing instruction is shown below:

```
(\d{1,5})+\s
```

This instruction says to look for a N followed by 1 to 5 numerical digits. The + indicates that the value must occur 1 or more times to match and the \s says that the NXXXX may be followed by a white space character, a tab, form-feed or carriage return.

The M or G command is parsed by the follow RegEx expression:

```
((\d{1,3})*\s*)
```

This expression say to look for an M or a G followed by 1 to 3 numerical digits. The \s* says that match may occur 0 or more times. The \s* indicates that white spaces may or may not be
Occur 0 or more times. The \s* indicates that white spaces may or may not be included after the M or G instruction.

The X location instruction is parsed with the following command:
```
*X[0-9]+*\d\d{0,3},*\d{0,4})"\n```

This instruction is similar to the ones above except that the X may or may not include a – or negative sign, 0 to 3 digits, a . or period may or may not be included followed by 0 to 4 additional digits.

More information is available on the Web at http://regex.oshervce.com

**NOTE:** Corruption of this line may disable the G Code Converter. Please make a backup of the original instruction before you make any modifications to it.

---

**ParseGroups**

**StartFile**

This value is optional. It may have any string value.

A typical value is the %, but may vary among different G Code Posts.

When this instruction is included, the G Code to Inform Converter will not process any commands until the StartFile Character is found.

**ThrowAwayLines**

Characters that are used in the G Code file to start comment lines. These lines will be ignored in the conversion process.

**ThrowAwayLinesRegEx**

This is the same as the ThrowAwayLines command but more that one character can be included. It is also a Regular Expression so can be used to catch more difficult sequences of characters that signify the start of a comment line.
The AngleSystem element is a required element that defines the angle system used in the G Code file. Values are YPR (Yaw, Pitch, and Roll) or Euler.

YPR is the Motoman and MotoSimEg standard

Euler angles is method of defining three rotations of an object about a point. It is somewhat of a standard in the Aerospace and Graphics industry. Euler Angles generally rotate about the ZYZ axes.

Example use:

```xml
<AngleSystem>YPR</AngleSystem>
```

This group of commands defines how the CNC A, B and C tool rotational axes are mapped into Robot Tool Angles Tx, Ty, Tz.

Normally the A CNC Angle is a Tool X rotation, the B CNC Angle rotates the Tool around the CNC Tool Y Axis and the C CNC Angle rotates the CNC about the CNC Tool Z axis.

The Z1, Y, Z2 Attributes come from the Euler angle naming convention. You can substitute or think of the Z1 as Tx, Y as Ty and Z2 as Tz robot angles.

For three degree of Freedom applications (X, Y, Z with a fixed tool orientation ) this section may be omitted from the Configuration File.

The following sub tags or commands Z1Negate, YNegate, and Z2Negate tell the robot if the A, B, and C angles should be negated when converting them to Tx, Ty, or Tz.

Example Use:
Example Use:
<FourToSixAxis Z1="A" Y="B" Z2="C">  
  <Z1Nega te>False</Z1Nega te>  
  <YNega te>False</YNega te>  
  <ZZNega te>True</ZZNega te>  
</FourToSixAxis>

FileConfiguration

Units

DefaultVelocity

ThreeDOF

Supplies the default orientation of the Robot Tool. This is the value that makes the last length of tool perpendicular to the part work plane.

ReferenceModel

ArmConfig

AlignXwithPath

Reserved for Future Development.

CircleFixUp

Machine tools and Motoman robots produce circles in very different ways.

G Code Circle Overview

G02: Circular motion, clockwise direction. There are two circular motion commands for cutting arcs, which are segments of circles, and full circles. Circular motion commands always move at the feedrate and are not rapid motion commands. The G02 command is the instruction for a clockwise circular motion. The arc will start at the tools current location and sweep around clockwise to the designated ending location at the programmed feedrate. There are several 'standard' methods to define the radius of the arc. These methods are discussed below, after the discussion on the counter-clockwise command.
G03: Circular motion, counter-clockwise direction. The G03 command is the instruction for counter-clockwise circular motion. The arc will start at the tools current location and sweep around counter-clockwise to the designated ending location at the programmed feedrate. There are several 'standard' methods to define the radius of an arc. These methods are discussed below, after the discussion on the counter-clockwise command.

Circular motion format, Arc Radius Definitions: A standard circular motion G-code instruction will always designate the arc's ending point in X, Y, and Z. However, there are several 'standard' formats for programming the arc's radius. Some controls support only one format, others support two or more formats. Refer to your machine's programming manual to determine which formats your CNC control supports.

The simplest method supported by many controls is to just include the radius value in the G-code line. The CNC control will do the calculations from there! For control's that support this format, the radius value is programmed by an 'R' word. In the following example a 0.5 inch radius will be machined (The Z value is modal in this example).

Note: If the CNC control can't calculate the defined radius, an error will occur at the machine.

G02 X1.0 Y1.0 R0.5 F10.

The most commonly supported format is the usage of I and J words to define the arc's center point location. Unfortunately, there are several different formats for defining the I and J locations!

The first I and J format some controls support, one that is pretty straightforward, is to program the arc center-point in absolute coordinates.

Program the arc's center-point location in the X axis with the 'I' word and the arc's center-point location in the Y axis with the 'J' word. This format, again one that makes a 0.5 inch radius, is written as follows:

G02 X1.75 Y1.25 I1.25 J0.75
**CircleJCenter**

When True, the I, J Vectors are

**DefaultToolNumber**

**DefaultNCPath**

**DCISimpleDrip**

This element is used with large files. Typically in Mold Creation. It causes the master job to add calls to load save and delete jobs in parallel with an executing subroutine.

Example: Excerpt from master job with DCISimpleDrip = True
(Causes the master job to load 3 sub routines.
'MotoSim Points Importer
'XML to JBI
LOADJ JOB:EXAM1001 JBI
LOADJ JOB:EXAM1002 JBI
LOADJ JOB:EXAM1003 JBI
CALL JOB:EXAM1001
SET 2 EXAM1004
DELETEJ JOB:EXAM1001 JBI
PSTART JOB:LOADJOB SUB2
CALL JOB:EXAM1002
SET 2 EXAM1005
DELETEJ JOB:EXAM1002 JBI
PSTART JOB:LOADJOB SUB2
CALL JOB:EXAM1003
SET 2 EXAM1006
DELETEJ JOB:EXAM1003 JBI
PSTART JOB:LOADJOB SUB2
CALL JOB:EXAM1004

**DCISimpleDripDVar**

Optional Element.

Used with DCISimpleDrip Element. DCISimpleDrip Element must be set to True before the G Code Converter will read this value.

**CannedVelocity**

**ExternalAxes**
<table>
<thead>
<tr>
<th>type</th>
<th>FileConfigurationType</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>content complex</td>
</tr>
<tr>
<td>children</td>
<td>RegExParseInt ParseGroups StartFile ThrowAwayLines ThrowAwayLinesRegEx ParseGroupsThrowAwayLines InformFileFormats AngleSystem MotionFunctions FourToSixAxis Units DefaultVelocity ThreeDOF ReferenceModel ArmConfig AlignXwithPath CircleFixUp CircleIJCenter DefaultToolNumber DefaultNCPath DCISimpleDrip DCISimpleDripDVar CannedVelocity ExternalAxes InformMotionFunctionGroup RotatingStationSetup BaseStation JobOutputOptions</td>
</tr>
</tbody>
</table>

source `<xs:element name="FileConfiguration" type="FileConfigurationType"/>`
element SetPosElmnt

<xs:element name="SetPosElmnt">
  <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="SourceNum" minOccurs="0"/>
      <xs:element ref="DestNum" minOccurs="0"/>
      <xs:element ref="DestIndex" minOccurs="0"/>
    </xs:choice>
  </xs:complexType>
</xs:element>

element SetVar
**SetVar** is an XML to JBI Instruction that allows the setting or Robot Variables. These Variables may be B, I, D or R variables.

P variables are set with SetPosElmnt.

This Element may have one of two forms. In form one, you specify the Variable Type and Index and provide the variable value with a "Const" element. In Form 2 you specify the source and destination variable types and indices.

```xml
<xs:element name="SetVar">
  <xs:annotation>
    <xs:documentation>SetVar is an XML to JBI Instruction that allows the setting or Robot Variables. These Variables may be B, I, D or R variables.

    P variables are set with SetPosElmnt.

    This Element may have one of two forms. In form one, you specify the Variable Type and Index and provide the variable value with a "Const" element.
    In Form 2 you specify the source and destination variable types and indices.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="Const" minOccurs="0"/>
      <xs:element ref="DestType" minOccurs="0"/>
      <xs:element ref="DestNum" minOccurs="0"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

**element ShiftFunctionType**
**Diagram**

Shift Function type is used to either setup a shift or to turn one on. Values can be ShiftOn or ShiftSetup.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>restriction of xs:string</td>
</tr>
<tr>
<td>Properties</td>
<td>content simple</td>
</tr>
<tr>
<td>Used by</td>
<td>complexType ShiftOffset</td>
</tr>
<tr>
<td>Facets</td>
<td>enumeration ShiftOn</td>
</tr>
<tr>
<td></td>
<td>enumeration ShiftSetup</td>
</tr>
<tr>
<td>Annotation</td>
<td>documentation</td>
</tr>
</tbody>
</table>

Shift Function type is used to either setup a shift or to turn one on. Values can be ShiftOn or ShiftSetup.

**Source**

```xml
<xs:element name="ShiftFunctionType">
  <xs:annotation>
    <xs:documentation>Shift Function type is used to either setup a shift or to turn one on. Values can be ShiftOn or ShiftSetup</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="ShiftOn"/>
      <xs:enumeration value="ShiftSetup"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**Element ShiftOffset**
### ShiftOffset

<table>
<thead>
<tr>
<th>type</th>
<th>ShiftOffset</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content complex</td>
</tr>
<tr>
<td>children</td>
<td>ShiftOffsetDVarIndex ShiftFunctionType ShiftSourceNum ShiftReferenceModel</td>
</tr>
<tr>
<td>used by</td>
<td>complexType INFORM</td>
</tr>
<tr>
<td>annotation</td>
<td>documentation</td>
</tr>
<tr>
<td></td>
<td>Mapps a G Code to a shift On or Shift Setup Function.</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;ShiftOffset&quot; type=&quot;ShiftOffset&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:annotation</a></td>
</tr>
<tr>
<td></td>
<td><a href="">xs:documentation</a>Mapps a G Code to a shift On or Shift Setup Function.&lt;/xs:documentation&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:annotation&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:element&gt;</td>
</tr>
</tbody>
</table>

### element ShiftOffsetDVarIndex

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Diagram]</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

| used by          | complexType ShiftOffset |
| annotation       | documentation |
|                  | Gives the D Variable that holds the additional Shift amount. |
element **ShiftOffsetTrueFalse**

**Diagram**

Used to turn on or Off a robot shift. Usually used with a G81 command.

**Type**

restriction of `xs:string`

**Properties**

content simple
default False

**Used by**

complexType **GCODE**

**Facets**

enumeration True
enumeration False

**Annotation**

documentation

Used to turn on or Off a robot shift. Usually used with a G81 command.

**Source**

```xml
<xs:element name="ShiftOffsetTrueFalse" default="False">
    <xs:annotation>
        <xs:documentation>Used to turn on or Off a robot shift. Usually used with a G81 command.</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="True"/>
            <xs:enumeration value="False"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

---

**element ShiftOn**

**Diagram**

Tells the robot the reference frame of the position. This can be:
- RF for Robot Frame,
- BF for Base Frame,
- TF for tool Frame,
- UFx for User Frame, or it can point to a model in the MotoSimEG Cell.

**Properties**

content complex
The `ShiftOn` element is used to specify the reference frame of the position. This can be:
- RF for Robot Frame,
- BF for Base Frame,
- TF for tool Frame,
- UFx for User Frame, or it can point to a model in the MotoSmEG Cell.

The `ShiftReferenceModel` element provides the user frame for the shift.

The `ShiftSourceNum` element represents the P Variable Index Number.
**element SourceNum**

<table>
<thead>
<tr>
<th>type</th>
<th>xs:integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td></td>
<td>default 1</td>
</tr>
<tr>
<td>used by</td>
<td>elements SetPosElmnt ShiftOn</td>
</tr>
<tr>
<td>source</td>
<td><code>&lt;xs:element name=&quot;SourceNum&quot; type=&quot;xs:integer&quot; default=&quot;1&quot;/&gt;</code></td>
</tr>
</tbody>
</table>

**element StartCannedCycle**

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>complexType GCODE</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration True</td>
</tr>
<tr>
<td></td>
<td>enumeration False</td>
</tr>
<tr>
<td>annotation</td>
<td>documentation</td>
</tr>
<tr>
<td></td>
<td>Starts the G Code Canned Cycle.</td>
</tr>
<tr>
<td>source</td>
<td><code>&lt;xs:element name=&quot;StartCannedCycle&quot;&gt;</code></td>
</tr>
</tbody>
</table>

**element StartFile**
element **StartFile**

<table>
<thead>
<tr>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple default %</td>
</tr>
<tr>
<td>used by</td>
<td>complexType FileConfigurationType</td>
</tr>
<tr>
<td>annotation</td>
<td>This value is optional. It may have any string value. A typical value is the %, but may vary among different G Code Posts. When this instruction is included, the G Code to Inform Converter will not process any commands until the StartFile Character is found.</td>
</tr>
</tbody>
</table>
| source        | `<xs:element name="StartFile" type="xs:string" default="%">  
<xs:annotation>  
  `<xs:documentation>This value is optional. It may have any string value. A typical value is the %, but may vary among different G Code Posts.  
  
When this instruction is included, the G Code to Inform Converter will not process any commands until the StartFile Character is found.</xs:documentation>  
</xs:annotation>  
</xs:element>` |

**element T**

<table>
<thead>
<tr>
<th>type</th>
<th>xs:integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>complexType ArmConfigType</td>
</tr>
<tr>
<td>source</td>
<td>`&lt;xs:element name=&quot;T&quot; type=&quot;xs:integer&quot;/&gt;'</td>
</tr>
</tbody>
</table>

element ThreeDOF
**ThreeDOFType**

Properties: complex

Attributes:
- **Rx**: xs:decimal, required
- **Ry**: xs:decimal, required
- **Rz**: derived by xs:decimal, required

Annotation documentation:
Supplies the default orientation of the Robot Tool. This is the value that makes the last length of tool perpendicular to the part work plane.

Source:
```xml
<xs:element name="ThreeDOF" type="ThreeDOFType">
  <xs:annotation>
    <xs:documentation>Supplies the default orientation of the Robot Tool. This is the value that makes the last length of tool perpendicular to the part work plane.</xs:documentation>
  </xs:annotation>
</xs:element>
```

**ThrowAwayLines**

Type: xs:string

Properties: simple

Used by: complexType **FileConfigurationType**

Annotation documentation:
Characters that are used in the G Code file to start comment lines. These lines will be ignored in the conversion process.
element **ThrowAwayLinesRegEx**

```xml
<xs:element name="ThrowAwayLinesRegEx" type="xs:string">
    <xs:annotation>
        <xs:documentation>This is the same as the ThrowAwayLines command but more that one character can be included. It is also a Regular Expression so can be used to catch more difficult sequences of characters that signify the start of a comment line.</xs:documentation>
    </xs:annotation>
</xs:element>
```

**Diagram:**

```
Timer
```

**Type:** restriction of `xs:decimal`

**Facets:**
- enumeration `0.5`
- enumeration `1.0`
annotation documentation
Creates a time delay Inform Function. Time delay is in seconds.

source
<xs:element name="Timer">
  <xs:annotation>
    <xs:documentation>Creates a time delay Inform Function. Time delay is in seconds.</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:decimal">
      <xs:enumeration value="0.5"/>
      <xs:enumeration value="1.0"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

element TimeUnits

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>complexType UnitsType</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration Minutes</td>
</tr>
<tr>
<td></td>
<td>enumeration Seconds</td>
</tr>
</tbody>
</table>

source
<xs:element name="TimeUnits">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Minutes"/>
      <xs:enumeration value="Seconds"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

element U

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>xs:integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>complexType ArmConfigType</td>
</tr>
</tbody>
</table>

source
<xs:element name="U" type="xs:integer"/>

element Velocity
<table>
<thead>
<tr>
<th>element</th>
<th>YNegate</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:string</td>
</tr>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>complexType FourToSixAxisType</td>
</tr>
</tbody>
</table>

### Velocity

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>complexType INFORM</td>
</tr>
<tr>
<td>facets</td>
<td>pattern (\d{0,4})</td>
</tr>
<tr>
<td>annotation</td>
<td>documentation</td>
</tr>
</tbody>
</table>

This element is used in the Position element. It gives the velocity which the robot will use to move from its current position to the specified position.

Note: If the Motion Type is Linear, the Velocity has a range of about 0 to 1600 millimeters/sec. If the Motion type is Joint the Velocity has a range from 0 to 100%.

```xml
<xs:element name="Velocity">
  <xs:annotation>
    <xs:documentation>This element is used in the Position element. It gives the velocity which the robot will use to move from its current position to the specified position.

    Note: If the Motion Type is Linear, the Velocity has a range of about 0 to 1600 millimeters/sec. If the Motion type is Joint the Velocity has a range from 0 to 100%.
  </xs:documentation>
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:pattern value="\d{0,4}\"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```
<table>
<thead>
<tr>
<th>facets</th>
<th>enumeration True</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>enumeration False</td>
</tr>
<tr>
<td>annotation documentation</td>
<td>When true, it causes the associated CNC Orientation Angle to be multiplied by -1.</td>
</tr>
</tbody>
</table>

```xml
<xs:element name="YNegate">
  <xs:annotation>
    <xs:documentation>When true, it causes the associated CNC Orientation Angle to be multiplied by -1.</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="True"/>
      <xs:enumeration value="False"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**element Z1Negate**

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of</th>
<th>xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>restriction of</td>
<td>xs:string</td>
</tr>
<tr>
<td></td>
<td>content</td>
<td>simple</td>
</tr>
<tr>
<td>used by</td>
<td>complexType</td>
<td>FourToSixAxisType</td>
</tr>
</tbody>
</table>

**annotation documentation**

When true, it causes the associated CNC Orientation Angle to be multiplied by -1.

```xml
<xs:element name="Z1Negate">
  <xs:annotation>
    <xs:documentation>When true, it causes the associated CNC Orientation Angle to be multiplied by -1.</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="True"/>
      <xs:enumeration value="False"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**element Z2Negate**

```xml
<xs:element name="Z2Negate">
  <xs:annotation>
    <xs:documentation>When true, it causes the associated CNC Orientation Angle to be multiplied by -1.</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="True"/>
      <xs:enumeration value="False"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```
<table>
<thead>
<tr>
<th>type</th>
<th>restriction of <code>xs:string</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>complexType <code>FourToSixAxisType</code></td>
</tr>
<tr>
<td>facets</td>
<td>enumeration True</td>
</tr>
<tr>
<td></td>
<td>enumeration False</td>
</tr>
<tr>
<td>annotation</td>
<td>documentation</td>
</tr>
<tr>
<td></td>
<td>When true, it causes the associated CNC Orientation Angle to be multiplied by -1.</td>
</tr>
</tbody>
</table>

```xml
<xs:element name="Z2Negate">
  <xs:annotation>
    <xs:documentation>When true, it causes the associated CNC Orientation Angle to be multiplied by -1.</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="True"/>
      <xs:enumeration value="False"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**complexType `ArmConfigType`**
used by element **FileConfigurationType/ArmConfig**

<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JobName</td>
<td>xs:string</td>
<td>required</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**annotation documentation**

Specifies the robot initial starting position and form. This position is specified in joint coordinates and will set the initial pose (elbow up/dow, etc.) of the robot.

This position can be set by moving the simulated robot to the part coordinate system 0,0,0 location and setting the individual axes to archive the desired pose. The values can be copied from MotoSimEG and pasted in the appropriate joint location.

Note: You can specify up to 21 unique robot configurations.

Note: The first set of Arm Configuration values are automatically selected.

**source**

```xml
<xs:complexType name="ArmConfigType">
  <xs:annotation>
    <xs:documentation>
      Specifies the robot initial starting position and form. This position is specified in joint coordinates and will set the initial pose (elbow up/dow, etc.) of the robot.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element ref="S"/>
    <xs:element ref="L"/>
  </xs:sequence>
</xs:complexType>
```
attribute ArmConfigType/@JobName

type xs:string

properties
  isRef 0
  use required

source <xs:attribute name="JobName" type="xs:string" use="required"/>

complexType BaseStationType

diagram

children BSMaximumLength BSMaximumCounts BSUserFrame BSRobot

used by element FileConfigurationType/BaseStation

source <xs:complexType name="BaseStationType">
  <xs:sequence>
    <xs:element name="BSMaximumLength" default="0.0">
      <xs:simpleType>
        <xs:restriction base="xs:float">
          <xs:pattern value="[+-]?\d+(\.\d+)?"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="BSMaximumCounts" default="0">
      <xs:simpleType>
        <xs:restriction base="xs:int">
          <xs:pattern value="[+-]?\d+"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="BSUserFrame">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="BSUFAxisAlignedWithBase" default="X">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="X"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="BSUserFrame"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
**Type** BaseStationType/BSMaximumLength

**Diagram**

```
<xs:element name="BSMaximumLength" default="0.0">
  <xs:restriction base="xs:float">
    <xs:pattern value="[+-]?\d+(\.+\d+)?">
      </xs:restriction>
  </xs:restriction>
</xs:element>
```

**Type** BaseStationType/BSMaximumCounts

**Diagram**

```
<xs:element name="BSMaximumCounts" default="0">
  <xs:restriction base="xs:int">
    <xs:pattern value="[+-]?\d+"/>
      </xs:restriction>
  </xs:restriction>
</xs:element>
```

**Type** BaseStationType/BSUserFrame

**Diagram**

```
<xs:element name="BSUserFrame">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="BSUFAxisAlignedWithBase"/>
      <xs:element name="BSUFNumber"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```
element BaseStationType/BSUserFrame/BSUFAxisAlignedWithBase

diagram

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="BSUFAxisAlignedWithBase" /></td>
</tr>
</tbody>
</table>

type restriction of xs:string

<table>
<thead>
<tr>
<th>properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>isRef</td>
</tr>
<tr>
<td>content</td>
</tr>
<tr>
<td>default</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>facets</th>
</tr>
</thead>
<tbody>
<tr>
<td>enumeration X</td>
</tr>
<tr>
<td>enumeration Y</td>
</tr>
<tr>
<td>enumeration Z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>source</th>
</tr>
</thead>
</table>
| <xs:element name="BSUFAxisAlignedWithBase" default="X">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="X"/>
      <xs:enumeration value="Y"/>
      <xs:enumeration value="Z"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element> |

---

element BaseStationType/BSUserFrame/BSUFNumber

diagram

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="BSUFNumber" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>source</th>
</tr>
</thead>
</table>
| <xs:element name="BSUFNumber" default="1">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:totalDigits value="2"/>
      <xs:fractionDigits value="0"/>
      <xs:maxInclusive value="25"/>
      <xs:minInclusive value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element> |
**type** | restriction of xs:integer  
---|---  
**properties** |  
| isRef | 0  
| content | simple  
| default | 1  
**facets** |  
| minInclusive | 1  
| maxInclusive | 25  
| totalDigits | 2  
| fractionDigits | 0  
**source** |  
```xml
<xs:element name="BSUFNumber" default="1">
    <xs:simpleType>
        <xs:restriction base="xs:integer">
            <xs:totalDigits value="2"/>
            <xs:fractionDigits value="0"/>
            <xs:maxInclusive value="25"/>
            <xs:minInclusive value="1"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

**element BaseStationType/BSRobot**  

**diagram**  

![Diagram](image)

**properties** |  
| isRef | 0  
| content | complex  
**children** |  
| BSRobotBaseAxis BSRobotMinBaseAxisValue BSRobotMaxAxisValue  
**source** |  
```xml
<xs:element name="BSRobot">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="BSRobotBaseAxis" default="X">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="X"/>
                        <xs:enumeration value="Y"/>
                        <xs:enumeration value="Z"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="BSRobotMinBaseAxisValue" default="0.0">
                <xs:simpleType>
                    <xs:restriction base="xs:float">
                        <xs:pattern value="[+-]?\d+(\.\d+)?"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="BSRobotMaxAxisValue" default="700.0">
                <xs:simpleType>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```
**element BaseStationType/BSRobot/BSRobotBaseAxis**

<table>
<thead>
<tr>
<th>diagram</th>
<th><a href="#">BSRobotBaseAxis</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of <code>xs:string</code></td>
</tr>
</tbody>
</table>
| properties | isRef 0  
|          | content `simple`  
|          | default `X`  |
| facets  | enumeration `X`  
|         | enumeration `Y`  
|         | enumeration `Z`  |
| source  | `<xs:element name="BSRobotBaseAxis" default="X">`  
|         | `<xs:restriction base="xs:string">`  
|         | `<xs:enumeration value="X"/>`  
|         | `<xs:enumeration value="Y"/>`  
|         | `<xs:enumeration value="Z"/>`  
|         | `</xs:restriction>`  
|         | `</xs:simpleType>`  
|         | `</xs:element>`  |

**element BaseStationType/BSRobot/BSRobotMinBaseAxisValue**

<table>
<thead>
<tr>
<th>diagram</th>
<th><a href="#">BSRobotMinBaseAxisValue</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of <code>xs:float</code></td>
</tr>
</tbody>
</table>
| properties | isRef 0  
|          | content `simple`  
|          | default `0.0`  |
| facets  | pattern `[+]?\d+(\.\d+)?` |
| source  | `<xs:element name="BSRobotMinBaseAxisValue" default="0.0">`  
|         | `<xs:restriction base="xs:float">`  
|         | `<xs:pattern value="[+]?\d+(\.\d+)?"/>`  
|         | `</xs:restriction>`  
|         | `</xs:simpleType>`  
|         | `</xs:element>`  |

**element BaseStationType/BSRobot/BSRobotMaxAxisValue**

<table>
<thead>
<tr>
<th>diagram</th>
<th><a href="#">BSRobotMaxAxisValue</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td></td>
</tr>
<tr>
<td>properties</td>
<td></td>
</tr>
<tr>
<td>facets</td>
<td></td>
</tr>
</tbody>
</table>
| source  | `<xs:element name="BSRobotMaxAxisValue" default=""/>`  
|         | `<xs:restriction base="xs:float">`  
|         | `<xs:pattern value="[+]?\d+(\.\d+)?"/>`  
|         | `</xs:restriction>`  
|         | `</xs:simpleType>`  
|         | `</xs:element>`  |
### BSRobotMaxAxisValue

**Type**: Restriction of `xs:float`

**Properties**
- `isRef`: 0
- `content`: Simple
- `default`: 700.0

**Facets**
- Pattern: `[+-]?d+(\.d+)?`

**Source**
```xml
<xs:element name="BSRobotMaxAxisValue" default="700.0">
  <xs:simpleType>
    <xs:restriction base="xs:float">
      <xs:pattern value="[+-]?d+(\.d+)?"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

---

### DefaultVelocityType

**Diagram**

```
DefaultVelocityType
  o---o
     ^  
  DefaultJointVelocity
     |
     o
  DefaultLinearVelocity
```

**Children**
- `DefaultJointVelocity`
- `DefaultLinearVelocity`

**Used by**
- `FileConfigurationType/DefaultVelocity`

**Source**
```xml
<xs:complexType name="DefaultVelocityType">
  <xs:sequence>
    <xs:element name="DefaultJointVelocity" type="xs:anySimpleType" minOccurs="0"/>
    <xs:element name="DefaultLinearVelocity" type="xs:anySimpleType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

---

### DefaultJointVelocity

**Diagram**

```
DefaultJointVelocity
```

**Type**: `xs:anySimpleType`

**Properties**
- `isRef`: 0
- `minOcc`: 0
- `maxOcc`: 1
- `content`: Simple

**Source**
```xml
<xs:element name="DefaultJointVelocity" type="xs:anySimpleType" minOccurs="0"/>
```

---

### DefaultLinearVelocity

**Diagram**

```
DefaultLinearVelocity
```

**Type**: `xs:anySimpleType`

**Source**
```xml
<xs:element name="DefaultLinearVelocity" type="xs:anySimpleType" minOccurs="0"/>
```
### complexType `ExternalAxesType`

This element is used to define an robot External Axis position if one exists in the robotic cell.

<table>
<thead>
<tr>
<th>properties</th>
<th>mixed 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>attributes</td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td>AxesType</td>
</tr>
<tr>
<td>annotation</td>
<td>documentation</td>
</tr>
<tr>
<td></td>
<td>This element is used to define an robot External Axis position if one exists in the robotic cell.</td>
</tr>
</tbody>
</table>

```xml
<xs:complexType name="ExternalAxesType" mixed="1">
  <xs:annotation>
    <xs:documentation>This element is used to define an robot External Axis position if one exists in the robotic cell.</xs:documentation>
  </xs:annotation>
  <xs:attribute name="AxesType"/>
  <xs:attribute name="AxesPulsePosition"/>
</xs:complexType>
```

### attribute `ExternalAxesType/@AxesType`

<table>
<thead>
<tr>
<th>properties</th>
<th>isRef 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td><code>&lt;xs:attribute name=&quot;AxesType&quot;/&gt;</code></td>
</tr>
</tbody>
</table>

### attribute `ExternalAxesType/@AxesPulsePosition`

<table>
<thead>
<tr>
<th>properties</th>
<th>isRef 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td><code>&lt;xs:attribute name=&quot;AxesPulsePosition&quot;/&gt;</code></td>
</tr>
</tbody>
</table>

### complexType `FileConfigurationType`
This is a required element.

This is a Regular Expression instruction. This is the instruction tells the G Code Converter which G Code Instructions to look for and the format of the G Code file.

This instruction is written in a way that most, if not all, G Code files can be converted without modifying this instruction. Many G Code fields are optional and may or may not appear in the G Code file. For example line Numbers may or may not be included in the G Code instruction.

Below is a RegExParseInt from one of the examples,

```xml
<RegExParseInt>(Nd\{1.5\})+\s)(([a-fA-F]|\d)(.3)(\d)(.4))s*([D][\d]{1,2}[0-9])s*([X][\d]{1,2}[0-9])s*([Y][\d]{1,2}[0-9])s*([Z][\d]{1,2}[0-9])s*([I][\d]{1,2}[0-9])s*([J][\d]{1,2}[0-9])s*([A][\d]{1,2}[0-9])s*([B][\d]{1,2}[0-9])s*([F][\d]{1,2}[0-9])s*([P][\d]{1,2}[0-9])</RegExParseInt>
```

It can be broken down into groups of instructions. The line number parsing instruction is shown below:

```xml
(Nd\{1.5\})+\s
```

This instruction says to look for a N followed by 1 to 5 numerical digits. The + indicates that the value must occur 1 or more times to match and the \s says that the NXXXX may be followed by a white space character, a tab, form-feed or carriage return.

The M or G command is parsed by the follow RegEx expression:

```xml
([MG][d\{1.3\}]+\s*)
```

This expression say to look for an M or a G followed by 1 to 3 numerical digits. The * says that match may occur 0 or more times. The \s* indicates that white spaces may or may not be included after the M or G instruction.
ParseGroupsThrowAwayLines

InformFileFormats

AngleSystem
The AngleSystem element is a required element that defines the angle system used in the G Code file. Values are YPR (Yaw, Pitch, and Roll) or Euler.

YPR is the Motoman and MotoSimEg standard.

Euler angles is method of defining three rotations of an object about a point. It is somewhat of a standard in the Aerospace and Graphics industry. Euler Angles generally rotate about the ZYX axes.

Example use:

\[ \text{\textlt{AngleSystem}\textgt{YPR}\textlt{\text\text\text\text\text\text}}} \]

MotionFunctions

FourToSixAxis
This group of commands defines how the CNC A, B and C tool rotational axes are mapped into Robot Tool Angles Tx, Ty, Tz.

Normally the A CNC Angle is a Tool X rotation, the B CNC Angle rotates the Tool around the CNC Tool Y Axis and the C CNC Angle rotates the CNC about the CNC Tool Z axis.

The Z1, Y, Z2 Attributes come form the Euler angle naming convention. You can substitute or think of the Z1 as Tx, Y as Ty and Z2 as Tz robot angles.

For three degree of Freedom applications (X, Y, Z with a fixed tool orientation) this section may be omitted from the Configuration File.

The following sub tags or commands Z1Negate, YNegate, and Z2Negate tell the robot if the A, B, and C angles should be negated when converting them to Tx, Ty, or Tz.

Example Use:

\[ \text{\textlt{FourToSixAxis Z1=\"A\" \text\\text\text\text\text\text}}} \text{Y=\"B\" Z2=\"C\"}}} \]
FileConfigurationType

This is the base element that defines all the setup and robot specific data that is used in the G Code to Inform Conversion. Here is where robot initial starting locations are defined and user frames. Dimensional units are also defined here. See the sub elements for more details.

ReferenceModel

ArmConfig

AlignXWithPath

CircleFixUp

Units

DefaultVelocity

ThreeDOF

Supplies the default orientation of the Robot Tool. This is the value that makes the last length of tool perpendicular to the part work plane.

G Code Circle Overview

G02: Circular motion, clockwise direction. There are two circular motion commands for cutting arcs, which are segments of circles, and full circles. Circular motion commands always move at the feedrate and are not rapid motion commands. The G02 command is the instruction for a clockwise circular motion. The arc will start at the tool's current location and sweep around clockwise to the designated ending location at the programmed feedrate. There are several 'standard' methods to define the radius of the arc. These methods are discussed below, after the discussion on the counter-clockwise command.

G03: Circular motion, counter-clockwise direction. The G03 command is the
The G03 command is the instruction for a counter-clockwise circular motion. The arc will start at the tool's current location and sweep counter-clockwise to the designated ending location at the programmed feedrate. There are several 'standard' methods to define the radius of an arc. These methods are discussed below, after the discussion on the counter-clockwise command.

Circular motion format. Arc Radius Definitions: A standard circular motion G-code instruction will always designate the arc's ending point in X, Y, and Z. However, there are several 'standard' formats for programming the arc's radius. Some controls support only one format, others support two or more formats. Refer to your machine's programming manual to determine which formats your CNC control supports. The simplest method supported by many controls is to just include the radius value in the G-code line. The CNC control will do the calculations from there! For control's that support this format, the radius value is programmed by an 'R' word. In the following example a 0.5 inch radius will be machined (The Z value is modal in this example).

Note: If the CNC control can't calculate the defined radius, an error will occur at the machine.

G02 X1.0 Y1.0 R0.5 F10.

The most commonly supported format is the usage of I and J words to define the arc's center point location. Unfortunately, there are several different formats for defining the I and J locations! The first I and J format some controls support, one that is pretty straightforward, is to program the arc center-point in absolute coordinates.

Program the arc's center-point location in the X axis with the 'I' word and the arc's center-point location in the Y axis with the 'J' word. This format, again one that makes a 0.5 inch radius, is written as follows:

G02 X1.75 Y1.25 I1.25 J0.75 F10.
CircleJCenter
When True, the I, J Vectors are

DefaultToolNumber

DefaultNCPPath

DCISimpleDrip
This element is used with large files. Typically in Mold Creation. It causes the master job to add calls to load save and delete jobs in parallel with an executing subroutine.
Example: Excerpt from master job with DCISimpleDrip = True (Causes the master job to load 3 sub routines.

DCISimpleDripDVar
Optional Element.
Used with DCISimpleDrip Element. DCISimpleDrip Element must be set to True before the G Code Converter will read this value.

CannedVelocity

ExternalAxes
ExternalAxes is used when the robot is mounted on an external axis; for example a
The base element that defines all the setup and robot specific data that is used in the G Code to Inform Conversion. Here is where robot initial starting locations are defined and user frames. Dimensional units are also defined here. See the sub elements for more details.
This is the base element that defines all the setup and robot specific data that is used in the G Code to inform conversion. Here is where robot initial starting locations are defined and user frames. Dimensional units are also defined here. See the sub elements for more details.

Normally the A CNC Angle is a Tool X rotation, the B CNC Angle rotates the Tool around the CNC Tool Y Axis and the C CNC Angle rotates the CNC about the CNC Tool Z axis.

The Z1, Y, Z2 Attributes come from the Euler angle naming convention. You can substitute or think of the Z1 as Tx, Y as Ty and Z2 as Tz robot angles.

For three degree of freedom applications (X, Y, Z with a fixed tool orientation) this section may be omitted from the Configuration File.

The following sub tags or commands Z1Negate, YNegate, and Z2Negate tell the robot if the A, B, and C angles should be negated when converting them to Tx, Ty, or Tz.

Example Use:

```xml
<FourToSixAxis Z1="A" Y="B" Z2="C">
  <Z1Negate>False</Z1Negate>
  <YNegate>False</YNegate>
  <Z2Negate>True</Z2Negate>
</FourToSixAxis>
```
<xs:documentation>Supplies the default orientation of the Robot Tool. This is the value that makes the last length of tool perpendicular to the part work plane.</xs:documentation>

<xs:element name="ReferenceModel">
   <xs:complexType mixed="true">
      <xs:attribute name="JobTypeMaster">
         <xs:simpleType>
            <xs:restriction base="xs:string">
               <xs:enumeration value="True"/>
            </xs:restriction>
         </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="JobTypeSub">
         <xs:simpleType>
            <xs:restriction base="xs:string">
               <xs:enumeration value="True"/>
            </xs:restriction>
         </xs:simpleType>
      </xs:attribute>
   </xs:complexType>
</xs:element>

<xs:element name="ArmConfig" type="ArmConfigType"/>

<xs:element ref="AlignXwithPath" minOccurs="0"/>

<xs:element ref="CircleFixUp"/>

<xs:element name="CircleIJCenter" default="False">
   <xs:annotation>
      <xs:documentation>When True, the I, J Vectors are </xs:documentation>
   </xs:annotation>
   <xs:simpleType>
      <xs:restriction base="xs:string">
         <xs:enumeration value="True"/>
         <xs:enumeration value="False"/>
      </xs:restriction>
   </xs:simpleType>
</xs:element>

<xs:element name="DefaultToolNumber" default="0" minOccurs="0">
   <xs:simpleType>
      <xs:restriction base="xs:integer">
         <xs:fractionDigits value="0"/>
         <xs:totalDigits value="2"/>
         <xs:minInclusive value="0"/>
         <xs:maxInclusive value="25"/>
      </xs:restriction>
   </xs:simpleType>
</xs:element>

<xs:element name="DefaultNCPath" type="xs:string" minOccurs="0"/>

<xs:element name="DCISimpleDrip" default="False" minOccurs="0">
   <xs:annotation>
      <xs:documentation>This element is used  with large files. Typically in Mold Creation. It causes the master job to add calls to load save and delete jobs in parallel with an executing subroutine.

Example: Excerpt from master job with DCISimpleDrip = True
(Causes the master job to load 3 sub routines.
'MotoSim Points Importer
</xs:documentation>
</xs:element>
\texttt{XML to JBI}

\texttt{LOADJ JOB:EXAM1001 JBI}
\texttt{LOADJ JOB:EXAM1002 JBI}
\texttt{LOADJ JOB:EXAM1003 JBI}
\texttt{CALL JOB:EXAM1001}
\texttt{SET 2 EXAM1004}
\texttt{DELETEJ JOB:EXAM1001 JBI}
\texttt{PSTART JOB:LOADJOB SUB2}
\texttt{CALL JOB:EXAM1002}
\texttt{SET 2 EXAM1005}
\texttt{DELETEJ JOB:EXAM1002 JBI}
\texttt{PSTART JOB:LOADJOB SUB2}
\texttt{CALL JOB:EXAM1003}
\texttt{SET 2 EXAM1006}
\texttt{DELETEJ JOB:EXAM1003 JBI}
\texttt{PSTART JOB:LOADJOB SUB2}
\texttt{CALL JOB:EXAM1004}

\texttt{\langle xs:documentation \rangle}
\texttt{\hspace{1cm} \langle xs:annotation \rangle}
\texttt{\hspace{2cm} \langle xs:simpleType \rangle}
\texttt{\hspace{3cm} \langle xs:restriction base="xs:string">}
\texttt{\hspace{4cm} \langle xs:enumeration value="True"/>}
\texttt{\hspace{4cm} \langle xs:enumeration value="False"/>}
\texttt{\hspace{3cm} \langle xs:restriction \rangle}
\texttt{\hspace{2cm} \langle xs:simpleType \rangle}
\texttt{\hspace{1cm} \langle xs:element \rangle}
\texttt{\hspace{2cm} \langle xs:element name="DCISimpleDripDVar" minOccurs="0">}
\texttt{\hspace{3cm} \langle xs:annotation \rangle}
\texttt{\hspace{4cm} \langle xs:documentation \rangle}Optional Element.
\texttt{\hspace{4cm} \langle xs:annotation \rangle}
\texttt{\hspace{3cm} \langle xs:element \rangle}
\texttt{\hspace{4cm} \langle xs:element name="DCISimpleDripDVar" minOccurs="0">}
\texttt{\hspace{5cm} \langle xs:annotation \rangle}
\texttt{\hspace{6cm} \langle xs:documentation \rangle}Optional Element.

\texttt{\langle xs:element name="DCISimpleDripDVar" minOccurs="0">}
\texttt{\langle xs:annotation \rangle}
\texttt{\langle xs:documentation \rangle}Optional Element.

\texttt{\langle xs:element name="DCISimpleDripDVar" minOccurs="0">}
\texttt{\langle xs:annotation \rangle}
\texttt{\langle xs:documentation \rangle}Optional Element.

\texttt{\langle xs:element name="DCISimpleDripDVar" minOccurs="0">}
\texttt{\langle xs:annotation \rangle}
\texttt{\langle xs:documentation \rangle}Optional Element.

\texttt{\langle xs:element ref="InformMotionFunctionGroup"/>}
element **FileConfigurationType/ParseGroups**

<table>
<thead>
<tr>
<th>type</th>
<th>ParseGroupsType</th>
</tr>
</thead>
</table>
| properties | isRef 0  
content complex |
attributes | Name | Type   | Use     | Default | Fixed | annotation
---|---|---|---|---|---|---
G1 | xs:string | optional
G2 | xs:string | optional
G3 | xs:string | optional
G4 | xs:string | optional
G5 | xs:string | optional
G6 | xs:string | optional
G7 | xs:string | optional
G8 | xs:string | optional
G9 | xs:string | optional
G10 | xs:string | optional
G11 | xs:string | optional
G12 | xs:string | optional
G13 | xs:string | optional
G14 | xs:string | optional
G15 | xs:string | optional
G16 | xs:string | optional
G17 | xs:string | optional

source | <xs:element name="ParseGroups" type="ParseGroupsType"/>

**element** FileConfigurationType/ParseGroupsThrowAwayLines

**source**
<xs:element name="ParseGroupsThrowAwayLines" type="ParseGroupsThrowAwayLinesType" minOccurs="0"/>

**element** FileConfigurationType/InformFileFormats
InformFileFormatsType (extension)

attributes

MasterandSubs

This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:

MasterName -- obsolete -- left in backwards compatibility - do not modify

MaxLines
Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

Example use:
InformFileFormats
 MasterandSubs="False"
 MasterName="gmaster"
 MaxLines="900"
 CallSub="M10"/

CallSub
Creates a Call Job instruction in the current robot job and when a specific M or G code is found. This function can be used for a number of functions.

Example use:
InformFileFormats
 MasterandSubs="False"
 MasterName="gmaster"
 MaxLines="750"
 CallSub="M10"/

ReturnSub
Creates a RET statement in the current Robot Job when a specific M or G code is

Example use:
InformFileFormats
 MasterandSubs="False"
 MasterName="gmaster"
 MaxLines="750"
 CallSub="M11"/

MasterName

MaxLines

CallSub

ReturnSub
<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>annotation documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MasterandSubs</td>
<td>xs:string</td>
<td>required</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:

- **MasterName** -- obsolete – left in backwards compatibility - do not modify
- **MaxLines**
  Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

Example use:
```
<InformFileFormats MasterandSubs="False"
  MasterName="gmaster"
  MaxLines="900"
  CallSub="M10"/>
```

- **CallSub**
  Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

Example use:
```
<InformFileFormats MasterandSubs="False"
  MasterName="gmaster"
  MaxLines="750"
  CallSub="M10"/>
```

- **ReturnSub**
  Creates a RET statement in the current Robot Job when a specific M or G code is
### Example use:
```
InformFileFormats
MasterAndSubs="False"
MasterName="gmaster"
MaxLines="750"
CallSub="M11"/
```

<table>
<thead>
<tr>
<th>MasterName</th>
<th>xs:string</th>
<th>optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxLines</td>
<td>xs:short</td>
<td>required</td>
</tr>
<tr>
<td>CallSub</td>
<td>xs:string</td>
<td>optional</td>
</tr>
<tr>
<td>ReturnSub</td>
<td>xs:string</td>
<td>optional</td>
</tr>
</tbody>
</table>

```
source <xs:element name="InformFileFormats">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="InformFileFormatsType"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

**element** `FileConfigurationType/MotionFunctions`

**type** `MotionFunctionsType`

**properties**
- `isRef`: 0
- `content`: complex

**children**
- `MotionFunction`

```
source <xs:element name="MotionFunctions" type="MotionFunctionsType"/>
```

**element** `FileConfigurationType/FourToSixAxis`
This group of commands defines how the CNC A, B and C tool rotational axes are mapped into Robot Tool Angles Tx, Ty, Tz.

Normally the A CNC Angle is a Tool X rotation, the B CNC Angle rotates the Tool around the CNC Tool Y Axis and the C CNC Angle rotates the CNC about the CNC Tool Z axis.

The Z1, Y, Z2 Attributes come from the Euler angle naming convention. You can substitute or think of the Z1 as Tx, Y as Ty and Z2 as Tz robot angles.

For three degree of Freedom applications (X, Y, Z with a fixed tool orientation) this section may be omitted from the Configuration File.

The following sub tags or commands Z1Negate, YNegate, and Z2Negate tell the robot if the A, B, and C angles should be negated when converting them to Tx, Ty, or Tz.

Example Use:

```xml
<FourToSixAxis Z1="A" Y="B" Z2="C">
  <Z1Negate>False</Z1Negate>
  <YNegate>False</YNegate>
  <Z2Negate>True</Z2Negate>
</FourToSixAxis>
```

| type          | FourToSixAxisType |
This group of commands defines how the CNC A, B and C tool rotational axes are mapped into Robot Tool Angles Tx, Ty, Tz. Normally the A CNC Angle is a Tool X rotation, the B CNC Angle rotates the Tool around the CNC Tool Y Axis and the C CNC Angle rotates the CNC about the CNC Tool Z axis.

The Z1, Y, Z2 Attributes come from the Euler angle naming convention. You can substitute or think of the Z1 as Tx, Y as Ty and Z2 as Tz robot angles.

For three degree of Freedom applications (X, Y, Z with a fixed tool orientation) this section may be omitted from the Configuration File.

The following sub tags or commands Z1Negate, YNegate, and Z2Negate tell the robot if the A, B, and C angles should be negated when converting them to Tx, Ty, or Tz.

Example Use:

```xml
<FourToSixAxis Z1="A" Y="B" Z2="C">
  <Z1Negate>False</Z1Negate>
  <YNegate>False</YNegate>
  <Z2Negate>True</Z2Negate>
</FourToSixAxis>
```
element FileConfigurationType/Units

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>UnitsType</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>content complex</td>
</tr>
<tr>
<td>children</td>
<td>LengthUnits AngleUnits TimeUnits</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;Units&quot; type=&quot;UnitsType&quot;/&gt;</td>
</tr>
</tbody>
</table>

element FileConfigurationType/DefaultVelocity

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>DefaultVelocityType</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>minOccurs 0</td>
</tr>
<tr>
<td></td>
<td>maxOccurs 1</td>
</tr>
<tr>
<td></td>
<td>content complex</td>
</tr>
<tr>
<td>children</td>
<td>DefaultJointVelocity DefaultLinearVelocity</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;DefaultVelocity&quot; type=&quot;DefaultVelocityType&quot; minOccurs=&quot;0&quot;/&gt;</td>
</tr>
</tbody>
</table>

element FileConfigurationType/ThreeDOF
### ThreeDOFType

**Type:** ThreeDOFType  
  
**Properties:**
- `isRef`: 0  
- `content`: complex  

**Used by:** complexType **INFORM**  

**Attributes:**
- **Name**: Rx  
  **Type**: xs:decimal  
  **Use**: required  
- **Name**: Ry  
  **Type**: xs:decimal  
  **Use**: required  
- **Name**: Rz  
  **Type**: derived by: xs:decimal  
  **Use**: required

**Annotation:**

Supplies the default orientation of the Robot Tool. This is the value that makes the last length of tool perpendicular to the part work plane.

**Source:**

```xml
<xs:element name="ThreeDOF" type="ThreeDOFType">
  <xs:annotation>
    <xs:documentation>
      Supplies the default orientation of the Robot Tool. This is the value that makes the last length of tool perpendicular to the part work plane.
    </xs:documentation>
  </xs:annotation>
</xs:element>
```

### FileConfigurationType/ReferenceModel

**Element:** FileConfigurationType/ReferenceModel  

**Diagram:**

![ReferenceModel Diagram](image)

**Properties:**
- `isRef`: 0  
- `content`: complex  
  - mixed: true  

**Used by:** complexType **INFORM**
<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JobTypeMaster</td>
<td>derived</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>JobTypeSub</td>
<td>derived</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

source
```xml
<xs:element name="ReferenceModel">
    <xs:complexType mixed="true">
        <xs:attribute name="JobTypeMaster">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="True"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="JobTypeSub">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="True"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
</xs:element>
```

attribute FileConfigurationType/ReferenceModel/@JobTypeMaster

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration True</td>
</tr>
</tbody>
</table>

source
```xml
<xs:attribute name="JobTypeMaster">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="True"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
```

attribute FileConfigurationType/ReferenceModel/@JobTypeSub

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration True</td>
</tr>
</tbody>
</table>

source
```xml
<xs:attribute name="JobTypeSub">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="True"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
```
element FileConfigurationType/ArmConfig

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>ArmConfigType</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>content complex</td>
</tr>
<tr>
<td>children</td>
<td>S L U R B T</td>
</tr>
<tr>
<td>attributes</td>
<td>Name  Type  Use  Default  Fixed  annotation</td>
</tr>
<tr>
<td></td>
<td>JobName xs:string required</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;ArmConfig&quot; type=&quot;ArmConfigType&quot;/&gt;</td>
</tr>
</tbody>
</table>

element FileConfigurationType/CircleIJCenter

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>content simple</td>
</tr>
<tr>
<td></td>
<td>default False</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration True</td>
</tr>
<tr>
<td></td>
<td>enumeration False</td>
</tr>
<tr>
<td>annotation</td>
<td>documentation</td>
</tr>
<tr>
<td></td>
<td>When True, the I, J Vectors are</td>
</tr>
</tbody>
</table>
**element FileConfigurationType/DefaultToolNumber**

- **Diagram**
  
- **Type**: restriction of `xs:integer`
- **Properties**:
  - `isRef`: 0
  - `minOcc`: 0
  - `maxOcc`: 1
  - `content`: simple
  - `default`: 0
- **Facets**:
  - `minInclusive`: 0
  - `maxInclusive`: 25
  - `totalDigits`: 2
  - `fractionDigits`: 0

**source**

```xml
<xs:element name="DefaultToolNumber" default="0" minOccurs="0" maxOccurs="1" type="xs:integer">  
  <xs:restriction base="xs:integer">  
    <xs:fractionDigits value="0"/>  
    <xs:totalDigits value="2"/>  
    <xs:minInclusive value="0"/>  
    <xs:maxInclusive value="25"/>  
  </xs:restriction>  
</xs:element>
```

**element FileConfigurationType/DefaultNCPath**

- **Diagram**
  
- **Type**: `xs:string`
- **Properties**:
  - `isRef`: 0
  - `minOcc`: 0
  - `maxOcc`: 1
  - `content`: simple
- **Source**

```xml
<xs:element name="DefaultNCPath" type="xs:string" minOccurs="0"/>  
```

**element FileConfigurationType/DCISimpleDrip**

**source**

```xml
<xs:element name="CircleIJCenter" default="False">  
  <xs:annotation>  
    <xs:documentation>When True, the I, J Vectors are</xs:documentation>  
  </xs:annotation>  
  <xs:simpleType>  
    <xs:restriction base="xs:string">  
      <xs:enumeration value="True"/>  
      <xs:enumeration value="False"/>  
    </xs:restriction>  
  </xs:simpleType>  
</xs:element>
```
<table>
<thead>
<tr>
<th>type</th>
<th>restriction of <code>xs:string</code></th>
</tr>
</thead>
</table>
| properties | isRef 0  
|         | minOcc 0  
|         | maxOcc 1  
|         | content simple  
|         | default False  |
| facets | enumeration True  
|        | enumeration False  |
| annotation | documentation |
|             | This element is used with large files. Typically in Mold Creation. It causes the master job to add calls to load save and delete jobs in parallel with an executing subroutine. |
|             | Example: Excerpt from master job with DCISimpleDrip = True  
|             | (Causes the master job to load 3 sub routines.  
|             | MotoSim Points Importer  
|             | XML to JBI  
|             | LOADJ JOB:EXAM1001 JBI  
|             | LOADJ JOB:EXAM1002 JBI  
|             | LOADJ JOB:EXAM1003 JBI  
|             | CALL JOB:EXAM1001  
|             | SET 2 EXAM1004  
|             | DELETEJ JOB:EXAM1001 JBI  
|             | PSTART JOB:LOADJOB SUB2  
|             | CALL JOB:EXAM1002  
|             | SET 2 EXAM1005  
|             | DELETEJ JOB:EXAM1002 JBI  
|             | PSTART JOB:LOADJOB SUB2  
|             | CALL JOB:EXAM1003  
|             | SET 2 EXAM1006  
|             | DELETEJ JOB:EXAM1003 JBI  
|             | PSTART JOB:LOADJOB SUB2  
|             | CALL JOB:EXAM1004  
|             | CALL JOB:EXAM1001  
|             | CALL JOB:EXAM1002 JBI  
|             | CALL JOB:EXAM1003 JBI  
|             | CALL JOB:EXAM1001  
|             | SET 2 EXAM1004  
|             | DELETEJ JOB:EXAM1001 JBI  
|             | PSTART JOB:LOADJOB SUB2  
|             | CALL JOB:EXAM1002  
|             | SET 2 EXAM1005  
|             | DELETEJ JOB:EXAM1002 JBI |
PSTART JOB:LOADJOB SUB2
CALL JOB:EXAM1003
SET 2 EXAM1006
DELETEJ JOB:EXAM1003 JBI
PSTART JOB:LOADJOB SUB2
CALL JOB:EXAM1004

source <xs:element name="DCISimpleDrip" default="False" minOccurs="0">
  <xs:annotation>
    <xs:documentation>This element is used with large files. Typically in Mold Creation. It causes the master job to add calls to load save and delete jobs in parallel with an executing subroutine.

    Example: Excerpt from master job with DCISimpleDrip = True
    (Causes the master job to load 3 sub routines.
    'MotoSim Points Importer
    'XML to JBI
    LOADJ JOB:EXAM1001 JBI
    LOADJ JOB:EXAM1002 JBI
    LOADJ JOB:EXAM1003 JBI
    CALL JOB:EXAM1001
    SET 2 EXAM1004
    DELETEJ JOB:EXAM1001 JBI
    PSTART JOB:LOADJOB SUB2
    CALL JOB:EXAM1002
    SET 2 EXAM1005
    DELETEJ JOB:EXAM1002 JBI
    PSTART JOB:LOADJOB SUB2
    CALL JOB:EXAM1003
    SET 2 EXAM1006
    DELETEJ JOB:EXAM1003 JBI
    PSTART JOB:LOADJOB SUB2
    CALL JOB:EXAM1004
    </xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="True"/>
      <xs:enumeration value="False"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

element FileConfigurationType/DCISimpleDripDVar

diagram

```
Optional Element.

Used with DCISimpleDrip Element. DCISimpleDrip Element must be set to True before the G Code Converter will read this value.
```

type restriction of xs:string
properties
  isRef 0
  minOccurs 0
  maxOccurs 1
  content simple

facets
  pattern [D]\d{0,4}

annotation
documentation
  Optional Element.
  Used with DCISimpleDrip Element. DCISimpleDrip Element must be set to True before the G Code Converter will read this value.

source
  <xs:element name="DCISimpleDripDVar" minOccurs="0">
    <xs:annotation>
      <xs:documentation>Optional Element.
      Used with DCISimpleDrip Element. DCISimpleDrip Element must be set to True before the G Code Converter will read this value. </xs:documentation>
    </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:pattern value="[D]\d{0,4}"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>

**element FileConfigurationType/CannedVelocity**

**diagram**

```
CannedVelocity: 0,\infty
```

**type** xs:integer

**properties**
  isRef 0
  minOccurs 0
  maxOccurs unbounded
  content simple

**source**
  <xs:element name="CannedVelocity" type="xs:integer" minOccurs="0" maxOccurs="unbounded"/>

**element FileConfigurationType/ExternalAxes**

**diagram**

```
- ExternalAxes
  - AxesPulsePosition
  - AxesType
```

- **attributes**
  - ExternalAxes is used when the robot is mounted on an external axis; for example a robot on a linear track. This element allows the G Code program to position the robot at a set position on the track.
  - Note: Dynamic external axes are not supported in this release of G Code.
ExternalAxes is used when the robot is mounted on an external axis; for example a robot on a linear track. This element allows the G Code program to position the robot at a set position on the track.

Note: Dynamic external axes are not supported in this release of G Code.

```
<xs:element name="ExternalAxes" default="True" minOccurs="0">
  <xs:annotation>
    <xs:documentation>ExternalAxes is used when the robot is mounted on an external axis; for example a robot on a linear track. This element allows the G Code program to position the robot at a set position on the track.
    
    Note: Dynamic external axes are not supported in this release of G Code.
  </xs:documentation>
  <xs:complexType mixed="1">
    <xs:attribute name="AxesPulsePosition" type="xs:string" default="0"/>
    <xs:attribute name="AxesType" type="xs:string" default="Base"/>
  </xs:complexType>
</xs:element>
```
### Rotating Station Setup

**Type:** RotatingStationSetupType  
**Properties:**  
- isRef: 0  
- minOccurs: 0  
- maxOccurs: 1  
- Content: complex  
**Children:**  
- RotationStationName  
- RotatingStationUserFrameNumber  
- PartCordinatesRotateWithStation  
- DefaultRotatingVelocity  
- DefaultRotatingMotionType  
- DefaultRotatingUnits  
- RotatingStationToCNCAngleMapping  
**Source:**  
```xml
<xs:element name="RotatingStationSetup" type="RotatingStationSetupType" minOccurs="0"/>
```

### Base Station

**Type:** BaseStationType  
**Properties:**  
- isRef: 0  
- minOccurs: 0  
- maxOccurs: 1  
- Content: complex  
**Children:**  
- BSMAXIMUMLENGTH  
- BSMAXIMUMCOUNTS  
- BSUSERFRAME  
- BSRobot  
**Source:**  
```xml
<xs:element name="BaseStation" type="BaseStationType" minOccurs="0"/>
```
complexType **FourToSixAxisType**

This group of commands defines how the CNC A, B and C tool rotational axes are mapped into Robot Tool Angles Tx, Ty, Tz.

Normally the A CNC Angle is a Tool X rotation, the B CNC Angle rotates the Tool around the CNC Tool Y Axis and the C CNC Angle rotates the CNC about the CNC Tool Z axis.

The Z1, Y, Z2 Attributes come form the Euler angle naming convention. You can substitute or think of the Z1 as Tx, Y as Ty and Z2 as Tz robot angles.

For three degree of Freedom applications (X, Y, Z with a fixed tool orientation ) this section may be omitted from the Configuration File.

The following sub tags or commands Z1Negate, YNegate, and Z2Negate tell the robot if the A, B, and C angles should be negated when converting them to Tx, Ty, or Tz.

Example Use:

```xml
<FourToSixAxis Z1="A" Y="B" Z2="C">
  <Z1Negate>False</Z1Negate>
  <YNegate>False</YNegate>
  <Z2Negate>True</Z2Negate>
</FourToSixAxis>
```

<table>
<thead>
<tr>
<th>children</th>
<th>Z1Negate YNegate Z2Negate</th>
</tr>
</thead>
<tbody>
<tr>
<td>used by</td>
<td>element <strong>FileConfigurationType/FourToSixAxis</strong></td>
</tr>
</tbody>
</table>
attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z1</td>
<td>derived by: xs: string</td>
<td>optional</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>derived by: xs: string</td>
<td>optional</td>
<td>B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z2</td>
<td>derived by: xs: string</td>
<td>optional</td>
<td>C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

annotation documentation

This group of commands defines how the CNC A, B and C tool rotational axes are mapped into Robot Tool Angles Tx, Ty, Tz.

Normally the A CNC Angle is a Tool X rotation, the B CNC Angle rotates the Tool around the CNC Tool Y Axis and the C CNC Angle rotates the CNC about the CNC Tool Z axis.

The Z1, Y, Z2 Attributes come form the Euler angle naming convention. You can substitute or think of the Z1 as Tx, Y as Ty and Z2 as Tz robot angles.

For three degree of Freedom applications (X, Y, Z with a fixed tool orientation ) this section may be omitted from the Configuration File.

The following sub tags or commands Z1Negate, YNegate, and Z2Negate tell the robot if the A, B, and C angles should be negated when converting them to Tx, Ty, or Tz.

Example Use:

```xml
<FourToSixAxis Z1="A" Y="B" Z2="C">
    <Z1Negate>False</Z1Negate>
    <YNegate>False</YNegate>
    <Z2Negate>True</Z2Negate>
</FourToSixAxis>
```

source

```xml
<xs:complexType name="FourToSixAxisType">
    <xs:annotation>
        <xs:documentation>
            This group of commands defines how the CNC A, B and C tool rotational axes are mapped into Robot Tool Angles Tx, Ty, Tz.
            Normally the A CNC Angle is a Tool X rotation, the B CNC Angle rotates the Tool around the CNC Tool Y Axis and the C CNC Angle rotates the CNC about the CNC Tool Z axis.
            The Z1, Y, Z2 Attributes come form the Euler angle naming convention. You can substitute or think of the Z1 as Tx, Y as Ty and Z2 as Tz robot angles.
            For three degree of Freedom applications (X, Y, Z with a fixed tool orientation ) this section may be omitted from the Configuration File.
            The following sub tags or commands Z1Negate, YNegate, and Z2Negate tell the robot if the A, B, and C angles should be negated when converting them to Tx, Ty, or Tz.
            Example Use:

            ```xml
                <FourToSixAxis Z1="A" Y="B" Z2="C">
                    <Z1Negate>False</Z1Negate>
                    <YNegate>False</YNegate>
                    <Z2Negate>True</Z2Negate>
                </FourToSixAxis>
            ```
        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element ref="Z1Negate"/>
        <xs:element ref="YNegate"/>
    </xs:sequence>
</xs:complexType>
```
attribute FourToSixAxisType/@Z1

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>default A</td>
</tr>
<tr>
<td></td>
<td>use optional</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration A</td>
</tr>
<tr>
<td></td>
<td>enumeration B</td>
</tr>
<tr>
<td></td>
<td>enumeration C</td>
</tr>
<tr>
<td></td>
<td>enumeration Empty</td>
</tr>
<tr>
<td></td>
<td>enumeration</td>
</tr>
</tbody>
</table>
attribute `FourToSixAxisType/@Y`

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of <code>xs:string</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>default B</td>
</tr>
<tr>
<td></td>
<td>use optional</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration A</td>
</tr>
<tr>
<td></td>
<td>enumeration B</td>
</tr>
<tr>
<td></td>
<td>enumeration C</td>
</tr>
<tr>
<td></td>
<td>enumeration Empty</td>
</tr>
<tr>
<td>source</td>
<td><code>&lt;xs:attribute name=&quot;Y&quot; use=&quot;optional&quot; default=&quot;B&quot;&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:restriction base=&quot;xs:string&quot;&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;A&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;B&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;C&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;Empty&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:restriction&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:attribute&gt;</td>
</tr>
</tbody>
</table>

attribute `FourToSixAxisType/@Z2`

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of <code>xs:string</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>default C</td>
</tr>
<tr>
<td></td>
<td>use optional</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration A</td>
</tr>
<tr>
<td></td>
<td>enumeration B</td>
</tr>
<tr>
<td></td>
<td>enumeration C</td>
</tr>
<tr>
<td></td>
<td>enumeration Empty</td>
</tr>
<tr>
<td>source</td>
<td><code>&lt;xs:attribute name=&quot;Z2&quot; use=&quot;optional&quot; default=&quot;C&quot;&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:restriction base=&quot;xs:string&quot;&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;A&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;B&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;C&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;Empty&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:restriction&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:attribute&gt;</td>
</tr>
</tbody>
</table>
complexType GCODE

This element lists a G Code Instruction that will be converted to Inform. It is used in the GT0INFORM element which also has a matching INFORM element. These instructions work together to complete the conversion.

Header
This instruction is used in the G Code Tag. It indicates if the G Code function will map into an Inform Header Instruction. Values may be True or False.

Description
This command is used to list all of the G Code functions that will cause robot motion. It is used by the G Code Converter during the G Code to Inform Conversion to parse all commands with multiple instructions and place the motion functions last, next to the XYZ position values.

isLiteralInform
When true, this command allows the XML commands in the corresponding Inform tag to be placed in the XML Intermediate file without an further processing.

ShiftOffsetTrueFalse
Used to turn on or off a robot shift. Usually used with a G81 command.

MotionGroup
When True, indicates that
This element lists a G Code Instruction that will be converted to Inform. It is used in the GTOInform element which also has a matching INFORM element. These instructions work together to complete the conversion.

```xml
<xs:complexType name="GCODE">
    <xs:annotation>
        <xs:documentation>This element lists a G Code Instruction that will be converted to Inform. It is used in the GTOInform element which also has a matching INFORM element. These instructions work together to complete the conversion.</xs:documentation>
    </xs:annotation>
    <xs:choice maxOccurs="unbounded">
        <xs:element ref="Header"/>
        <xs:element ref="Description"/>
        <xs:element ref="MotionFunction"/>
        <xs:element ref="isLiteralInform"/>
        <xs:element ref="ShiftOffsetTrueFalse" minOccurs="0"/>
        <xs:element ref="MotionGroup" minOccurs="0"/>
        <xs:element ref="EndCannedCycle" minOccurs="0"/>
        <xs:element ref="StartCannedCycle" minOccurs="0"/>
    </xs:choice>
    <xs:attribute name="func" type="xs:anySimpleType" use="optional"/>
</xs:complexType>
```
This element controls how one G Code Functions is converted into a single or group of Robot Inform language commands. It has a set of sub functions: GCODE and INFORM that are required to convert one instruction.

```xml
<xs:complexType name="GCodeToInform">
  <xs:annotation>
    <xs:documentation>This element controls how one G Code Functions is converted into a single or group of Robot Inform language commands. It has a set of sub functions: GCODE and INFORM that are required to convert one instruction.</xs:documentation>
  </xs:annotation>
  <xs:sequence maxOccurs="2">
    <xs:element ref="GCODE"/>
    <xs:element ref="INFORM"/>
  </xs:sequence>
</xs:complexType>
```

complexType INFORM

```xml
<xs:complexType name="MotionType">
  <xs:annotation>
    <xs:documentation>MotionType allow the user to specify the type of robotic motion that will be used to send the robot to a position. The MotionType can be Circular, Joint, or Linear. Circular Motion will move the robot in a circular arc. Joint motion will time coordinate all robot axes and Linear will move the robot in a straight line.</xs:documentation>
  </xs:annotation>
</xs:complexType>
```

```xml
<xs:complexType name="OutputOn">
  <xs:annotation>
    <xs:documentation>This command is used to map a G or M Code into a robot Output On Command.</xs:documentation>
  </xs:annotation>
</xs:complexType>
```

```xml
<xs:complexType name="Timer">
  <xs:annotation>
    <xs:documentation>Create a time delay</xs:documentation>
  </xs:annotation>
</xs:complexType>
```
Timer

Creates a time delay
Inform Function. Time
delay is in seconds.

ThreeDOF

Supplies the default
orientation of the Robot
Tool. This is the value that
makes the last length of tool
perpendicular to the part
work plane.

LengthUnits

ReferenceModel

OutputOff

0...∞

Turns Off a robot Output
whos number is specified in
the OutputOff Tag.

GenInform

GenInform is an
abbreviation for General
Inform Any valid Inform
command can be used inside
the GenInform Command
but it is the responsibility of
the user to know the correct
syntax. This is a pass
through command that is not
syntax checked.

This command allows users
to use any map an G Code
Code command into any
Inform Instruction.
It is used in the
GTOINFORM— INFORM
section of the Configuration
File to map a G Code
Instruction into a Robot
Inform Instruction.

Example:
The following example maps
a G90 Instruction into a Shift
Off Inform instruction;

<GTOINFORM>
  <GCODE
    func="G90"/>
  <Header>False</Header>
  <Description>Linear
    Motion</Description>
  <isLiteralInform>True</isLit-
eralInform>
  <MotionFunction>False</M
    otionFunction>
  </GCODE>
  <INFORM>
<INFORM>

This instruction is used with in the GCODETOINFORM Instruction. It defines the XML based instructions that are sent to the Points Importer to convert a G Code Instruction into an Inform Instruction.

<CallJob>
This is an XML to Inform Instruction. Use this element to cause the robot to call a job subroutine.

This instruction will appear in the intermediate XML file that the G Code Converter creates during the conversion process. It will produce an Inform Instruction in the robot JBI with

Example use:
<CallJob>Pickup</CallJob>
Example output JBI: CALL JOB:Pickup
</CallJob>

<ShiftOffset>
Maps a G Code to a shift On or Shift Setup Function.

<Velocity>
This element is used in the Position element. It gives the velocity which the robot will use to move from its current position to the specified position.

Note: If the Motion Type is Linear, the Velocity has a range of about 0 to 1600 millimeters/sec. If the Motion type is Joint the Velocity has a range from 0 to 100%.

<PosLevel>
Position Level tells the controller how close to come to the programmed location. This is used to allow the robot to round or smooth direction changes. Level 0 makes the robot most accurate. Level 5 makes the smoothest path.

<MotionFunction>
This command is used to list all of the G Code functions that will cause robot motion. It is used by the G Code Converter during the G Code to Inform Conversion to parse all commands with multiple instructions and
Code to Inform Conversion to parse all commands with multiple instructions and place the motion functions last, next to the XYZ position values.

**ShiftSourceNum**

P Variable Index Number

**SetVar**

SetVar is an XML to JBI Instruction that allows the setting or Robot Variables. These Variables may be B, I, D or R variables.

P variables are set with SetPosElmnt.

This Element may have one of two forms. In form one, you specify the Variable Type and Index and provide the variable value with a "Const" element. In Form 2 you specify the source and destination variable types and indices.

**SetPosElmnt**

**ShiftOn**

Tells the robot the reference frame of the position. This can be:

- RF for Robot Frame,
- BF for Base Frame,
- TF for tool Frame,
- UFx for User Frame, or it can point to a model in the MotoSmEG Cell.

**InformMotionFunctionGroup**

This command allows the user to create more than one Motion Function for a single G Code Command.

Alternatively the user can call a job and the job can have multiple motion positions.

Example:

```xml
<InformMotionFunctionGroup
MotionType="Linear" ZValue="Z"
PosLevel="0"
CannedVelocityRegister="1"/>
<InformMotionFunctionGroup
MotionType="Inc" ZValue="Z"
PosLevel="0" PosVar="121"
FillWithDVar="1" ExtraOffsetDVar="2"
RefFrame="Tool"/>
<InformMotionFunctionGroup
MotionType="Inc" ZValue="R"
PosLevel="0"
CannedVelocityRegister="1"
PosVar="122" FillWithDVar="1"
ExtraOffsetDVar="5"/>
```
<table>
<thead>
<tr>
<th>children</th>
<th>MotionType OutputOn Timer ThreeDOF LengthUnits ReferenceModel OutputOff GenInform CallJob ShiftOffset Velocity PosLevel MotionFunction ShiftSourceNum SetVar SetPosElmnt ShiftOn InformMotionFunctionGroup RotatingStationPosition Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>used by</td>
<td>element INFORM</td>
</tr>
<tr>
<td>annotation documentation</td>
<td>This instruction is used within the GCODETOINFORM Instruction. It defines the XML based instructions that are sent to the Points Importer to convert a G Code Instruction into an Inform Instruction.</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:complexType name=&quot;INFORM&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:annotation</a></td>
</tr>
<tr>
<td></td>
<td><a href="">xs:documentation</a>This instruction is used with in the GCODETOINFORM Instruction. It defines the XML based instructions that are sent to the Points Importer to convert a G Code Instruction into an Inform Instruction.&lt;/xs:documentation&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:annotation&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:choice maxOccurs=&quot;unbounded&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element ref=&quot;MotionType&quot; minOccurs=&quot;0&quot; maxOccurs=&quot;unbounded&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element ref=&quot;OutputOn&quot; minOccurs=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element ref=&quot;Timer&quot; minOccurs=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element ref=&quot;ThreeDOF&quot; minOccurs=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element ref=&quot;LengthUnits&quot; minOccurs=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element ref=&quot;ReferenceModel&quot; minOccurs=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element ref=&quot;OutputOff&quot; minOccurs=&quot;0&quot; maxOccurs=&quot;unbounded&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element ref=&quot;GenInform&quot; minOccurs=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element ref=&quot;CallJob&quot; minOccurs=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element ref=&quot;ShiftOffset&quot; minOccurs=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element ref=&quot;Velocity&quot; minOccurs=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element ref=&quot;PosLevel&quot; minOccurs=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element ref=&quot;MotionFunction&quot; minOccurs=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element ref=&quot;ShiftSourceNum&quot; minOccurs=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element ref=&quot;SetVar&quot; minOccurs=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element ref=&quot;SetPosElmnt&quot; minOccurs=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element ref=&quot;ShiftOn&quot; minOccurs=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element ref=&quot;InformMotionFunctionGroup&quot; minOccurs=&quot;0&quot; maxOccurs=&quot;9&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element name=&quot;RotatingStationPosition&quot; type=&quot;RotatingStationJobDataPositionType&quot; minOccurs=&quot;0&quot; maxOccurs=&quot;4&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element name=&quot;Position&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:choice&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:complexType&gt;</td>
</tr>
</tbody>
</table>

element INFORM/RotatingStationPosition
type RotatingStationJobDataPositionType

properties
  isRef 0
  minOccurs 0
  maxOccurs 4
  content complex

children
  RotationStationName
  PositionUnits
  PositionValue
  RotatingVelocity
  RotatingMotionType

source

<xs:element name="RotatingStationPosition" type="RotatingStationJobDataPositionType" minOccurs="0" maxOccurs="4"/>

element INFORM/Position

properties isRef 0

source

<xs:element name="Position"/>

complexType InformFileFormatsType

element MasterandSubs

attributes

This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:

- MasterName -- obsolete -- left in backwards compatibility - do not modify
- MaxLines
  Set from 100 to 950.
  Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

Example use:
**InformFileFormatsType**

This element controls the form of the robot jobs that are created. The following Attributes are used:

- **MasterName** – obsolete – left in backwards compatibility - do not modify
- **MaxLines**
  Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

Example use:
```xml
<InformFileFormats
  MasterandSubs="False"
  MasterName="gmaster"
  MaxLines="900"
  CallSub="M10"/>
```

- **CallSub**
  Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

Example use:
```xml
<InformFileFormats
  MasterandSubs="False"
  MasterName="gmaster"
  MaxLines="750"
  CallSub="M10"/>
```

- **ReturnSub**
  Creates a RET statement in the current Robot Job when a specific M or G code is

Example use:
```xml
<InformFileFormats
  MasterandSubs="False"
  MasterName="gmaster"
  MaxLines="750"
  CallSub="M11"/>
```

---

Used by element **FileConfigurationType/InformFileFormats**
This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:

MasterName -- obsolete – left in backwards compatibility - do not modify

MaxLines
Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

Example use:
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="900"
CallSub="M10"/

CallSub
Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

Example use:
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="750"
CallSub="M10"/

ReturnSub
Creates a RET statement in the current Robot Job when a specific M or G code is

Example use:
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="750"
CallSub="M11"/
This element controls the form of the robot jobs that are created. The following Attributes are used:

**MasterName** -- obsolete – left in backwards compatibility - do not modify

**MaxLines**
Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="900" CallSub="M10"/

**CallSub**
Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M10"/

**ReturnSub**
Creates a RET statement in the current Robot Job when a specific M or G code is

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M11"/

This element is used to control the form of the Robot Inform jobs that are
InformFileFormatsType

<table>
<thead>
<tr>
<th>attribute</th>
<th>InformFileFormatsType/@MasterandSubs</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>xs:string</td>
</tr>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>use required</td>
</tr>
</tbody>
</table>

- **MasterandSubs**
  - type: xs:string
  - properties: isRef 0, use required
  - annotation: documentation
  
  This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:

  - **MasterName** — obsolete — left in backwards compatibility - do not modify
  
  - **MaxLines**
    - Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

  - Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="900" CallSub="M10"/

- **CallSub**
  - Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

  - Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M10"/

- **ReturnSub**
  - Creates a RET statement in the current Robot Job when a specific M or G code is

  - Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M11"/
Creates a RET statement in the current Robot Job when a specific M or G code is
Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M11"/

source <xs:attribute name="MasterandSubs" type="xs:string" use="required">
  <xs:documentation>This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:

  MasterName  -- obsolete – left in backwards compatibility - do not modify

  MaxLines
  Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

  Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="900" CallSub="M10"/

  CallSub
  Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

  Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M10"/

  ReturnSub
  Creates a RET statement in the current Robot Job when a specific M or G code is

  Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M11"/
</xs:documentation>
</xs:annotation>
</xs:attribute>

attribute InformFileFormatsType/@MasterName

<table>
<thead>
<tr>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0  use optional</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:attribute name=&quot;MasterName&quot; type=&quot;xs:string&quot; use=&quot;optional&quot;/&gt;</td>
</tr>
</tbody>
</table>

attribute InformFileFormatsType/@MaxLines

<table>
<thead>
<tr>
<th>type</th>
<th>xs:short</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0  use required</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:attribute name=&quot;MaxLines&quot; type=&quot;xs:short&quot; use=&quot;required&quot;/&gt;</td>
</tr>
</tbody>
</table>

attribute InformFileFormatsType/@CallSub

<table>
<thead>
<tr>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>&lt;xs:attribute name=&quot;CallSub&quot; type=&quot;xs:string&quot; use=&quot;optional&quot;/&gt;</td>
</tr>
</tbody>
</table>
| properties | isRef 0  
|            | use optional |

source  

```
<xs:attribute name="CallSub" type="xs:string" use="optional"/>
```

attribute InformFileFormatsType/@ReturnSub

| properties | isRef 0  
|            | use optional |

source  

```
<xs:attribute name="ReturnSub" type="xs:string" use="optional"/>
```

complexType InformMotionFunctionGroupType

diagram

![Diagram](image)

This command allows the user to create more than one Motion Function for a single G Code Command.

Alternatively the user can call a job and the job can have multiple motion positions.

Example:

```
<InformMotionFunctionGroup
  MotionType="Linear" ZValue="Z"
  PosLevel="0" CannedVelocityRegister="1"/>
```

```
<InformMotionFunctionGroup
  MotionType="Inc" ZValue="Z" PosLevels="0"
  PosVar="121" FillWithDVar="1"
  ExtraOffsetDVar="2" RefFrame="Tool"/>
```

```
<InformMotionFunctionGroup
  MotionType="Inc" ZValue="R" PosLevels="0"
  CannedVelocityRegister="1" PosVar="122"
  FillWithDVar="1" ExtraOffsetDVar="2"
  RefFrame="Tool"/>
```

| type | extension of xs:anySimpleType |
|      |                               |

| properties | base xs:anySimpleType |
|            |                       |

| used by | element InformMotionFunctionGroup |
|         |                                   |
This command allows the user to create more than one Motion Function for a single G Code Command. Alternatively the user can call a job and the job can have multiple motion positions.

Example:
```
<InformMotionFunctionGroup MotionType="Linear" ZValue="Z" PosLevel="0" CannedVelocityRegister="1"/>
<InformMotionFunctionGroup MotionType="Inc" ZValue="Z" PosLevel="0" PosVar="121" FillWithDVar="1" ExtraOffsetDVar="2" RefFrame="Tool"/>
<InformMotionFunctionGroup MotionType="Inc" ZValue="R" PosLevel="0" CannedVelocityRegister="1" PosVar="122" FillWithDVar="1" ExtraOffsetDVar="2" RefFrame="Tool"/>
```
attribute InformMotionFunctionGroupType/@MotionType

type restriction of xs:string
**properties**

- isRef: 0
- default: Linear
  - use: optional

**facets**

- enumeration: Linear
- enumeration: Inc

**source**

```xml
<xs:attribute name="MotionType" use="optional" default="Linear">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Linear"/>
      <xs:enumeration value="Inc"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

**attribute** `InformMotionFunctionGroupType/@ZValue`

**type**

- restriction of `xs:string`

**properties**

- isRef: 0
- default: Z
  - use: optional

**facets**

- enumeration: Z
- enumeration: R

**source**

```xml
<xs:attribute name="ZValue" use="optional" default="Z">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Z"/>
      <xs:enumeration value="R"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

**attribute** `InformMotionFunctionGroupType/@PosLevel`

**type**

- restriction of `xs:integer`

**properties**

- isRef: 0
- default: 0
  - use: optional

**facets**

- enumeration: 0
- enumeration: 1
- enumeration: 2
- enumeration: 3
- enumeration: 4
- enumeration: 5
### attribute InformMotionFunctionGroupType/@CannedVelocityRegister

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>default 1</td>
</tr>
<tr>
<td></td>
<td>use optional</td>
</tr>
<tr>
<td>facets</td>
<td>minInclusive 1</td>
</tr>
<tr>
<td></td>
<td>maxInclusive 125</td>
</tr>
</tbody>
</table>

**source**

```xml
<xs:attribute name="CannedVelocityRegister" use="optional" default="1">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="125"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

### attribute InformMotionFunctionGroupType/@PosVar

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>default 1</td>
</tr>
<tr>
<td></td>
<td>use optional</td>
</tr>
<tr>
<td>facets</td>
<td>minInclusive 1</td>
</tr>
<tr>
<td></td>
<td>maxInclusive 125</td>
</tr>
</tbody>
</table>

**source**

```xml
<xs:attribute name="PosVar" use="optional" default="1">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="125"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

### attribute InformMotionFunctionGroupType/@FillWithDVar

| type            | restriction of xs:integer |

**source**

```xml
<xs:attribute name="FillWithDVar" use="optional" default="0">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:enumeration value="0"/>
      <xs:enumeration value="1"/>
      <xs:enumeration value="2"/>
      <xs:enumeration value="3"/>
      <xs:enumeration value="4"/>
      <xs:enumeration value="5"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```
### FillWithDVar

- **Type**: restriction of `xs:integer`
- **Properties**:
  - `isRef 0`
  - `default 1`
  - `use optional`
- **Facets**: `minInclusive 1`, `maxInclusive 125`
- **Source**
  ```xml
  <xs:attribute name="FillWithDVar" use="optional" default="1">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="125"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  ```

### ExtraOffsetDVar

- **Type**: restriction of `xs:integer`
- **Properties**:
  - `isRef 0`
  - `default 2`
  - `use optional`
- **Facets**: `minInclusive 1`, `maxInclusive 125`
- **Source**
  ```xml
  <xs:attribute name="ExtraOffsetDVar" use="optional" default="2">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="125"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  ```

### InputPositionDataType

- **ComplexType**
- **Diagram**
- **Children**:
  - `InputPositionDataPulse`
  - `InputPositionDataTypeRectan`
- **Used by**:
  - element `InputPositionData`
<xs:complexType name="InputPositionDataType">
  <xs:choice>
    <xs:element name="InputPositionDataPulse" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="PositionConfigurationPulse" type="PositionConfigurationPulseType"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="InputPositionDataTypeRectan" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="PositionConfiguration" type="PositionConfigurationType"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
element `InputPositionDataType/InputPositionDataTypeRectan`

```
<xs:element name="InputPositionDataTypeRectan" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="PositionConfiguration" type="PositionConfigurationType"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

```
<xs:complexType>
    <xs:sequence>
        <xs:element name="PositionConfiguration" type="PositionConfigurationType"/>
    </xs:sequence>
</xs:complexType>
```

```
<xs:element name="PositionConfiguration" type="PositionConfigurationType"/>
```

```
<xs:complexType>
    <xs:sequence>
        <xs:element name="PositionConfiguration" type="PositionConfigurationType"/>
        <xs:element name="RobotFrame"/>
        <xs:element name="BaseFrame"/>
        <xs:element name="UserFrame"/>
    </xs:sequence>
</xs:complexType>
```

complexType `JobOutputOptionsType`

```
<xs:element name="JobOutputOptions" type="JobOutputOptionsType"/>
```

```
<xs:complexType>
    <xs:sequence>
        <xs:element name="JobType"/>
    </xs:sequence>
</xs:complexType>
```

```
<xs:element name="JobOutputOptions" type="JobOutputOptionsType"/>
```
<xs:complexType name="JobOutputOptionsType">
  <xs:all>
    <xs:element name="JobType">
      <xs:complexType>
        <xs:choice>
          <xs:element name="PulseJobType" default="True" minOccurs="0">
            <xs:complexType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="True"/>
                <xs:enumeration value="False"/>
              </xs:restriction>
            </xs:complexType>
          </xs:element>
          <xs:element name="RectanJobType" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="PositionConfiguration" type="PositionConfigurationType"/>
                <xs:element name="RobotPosture">
                  <xs:complexType>
                    <xs:all>
                      <xs:element name="FlipPosture" default="0">
                        <xs:complexType>
                          <xs:restriction base="xs:integer">
                            <xs:enumeration value="0"/>
                            <xs:enumeration value="1"/>
                          </xs:restriction>
                        </xs:complexType>
                      </xs:element>
                      <xs:element name="UpperPosture" default="0">
                        <xs:complexType>
                          <xs:restriction base="xs:integer">
                            <xs:enumeration value="0"/>
                            <xs:enumeration value="1"/>
                          </xs:restriction>
                        </xs:complexType>
                      </xs:element>
                      <xs:element name="FrontPosture" default="0">
                        <xs:complexType>
                          <xs:restriction base="xs:integer">
                            <xs:enumeration value="0"/>
                            <xs:enumeration value="1"/>
                          </xs:restriction>
                        </xs:complexType>
                      </xs:element>
                      <xs:element name="RPosture" default="0">
                        <xs:complexType>
                          <xs:restriction base="xs:integer">
                            <xs:enumeration value="0"/>
                            <xs:enumeration value="1"/>
                          </xs:restriction>
                        </xs:complexType>
                      </xs:element>
                      <xs:element name="TPosture" default="0">
                        <xs:complexType>
                          <xs:restriction base="xs:integer">
                            <xs:enumeration value="0"/>
                            <xs:enumeration value="1"/>
                          </xs:restriction>
                        </xs:complexType>
                      </xs:element>
                    </xs:all>
                  </xs:complexType>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:all>
</xs:complexType>
**element** JobOutputOptionsType/JobType

**properties**
- isRef: 0
- content: complex

**children**
- PulseJobType
- RectanJobType

**source**
```xml
<xs:element name="JobType">
  <xs:complexType>
    <xs:choice>
      <xs:element name="PulseJobType" default="True" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="True"/>
            <xs:enumeration value="False"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="RectanJobType" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="PositionConfiguration" type="PositionConfigurationType"/>
            <xs:element name="RobotPosture"/>
          </xs:complexType>
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>
</xs:element>
```
<xs:element name="UpperPosture" default="0">
 <xs:simpleType>
  <xs:restriction base="xs:integer">
   <xs:enumeration value="0"/>
   <xs:enumeration value="1"/>
  </xs:restriction>
 </xs:simpleType>
</xs:element>

<xs:element name="FrontPosture" default="0">
 <xs:simpleType>
  <xs:restriction base="xs:integer">
   <xs:enumeration value="0"/>
   <xs:enumeration value="1"/>
  </xs:restriction>
 </xs:simpleType>
</xs:element>

<xs:element name="RPosture" default="0">
 <xs:simpleType>
  <xs:restriction base="xs:integer">
   <xs:enumeration value="0"/>
   <xs:enumeration value="1"/>
  </xs:restriction>
 </xs:simpleType>
</xs:element>

<xs:element name="TPosture" default="0">
 <xs:simpleType>
  <xs:restriction base="xs:integer">
   <xs:enumeration value="0"/>
   <xs:enumeration value="1"/>
  </xs:restriction>
 </xs:simpleType>
</xs:element>

</xs:all>
</xs:complexType>
</xs:element>

<table>
<thead>
<tr>
<th>element</th>
<th>JobOutputOptionsType/JobType/PulseJobType</th>
</tr>
</thead>
<tbody>
<tr>
<td>diagram</td>
<td><img src="image" alt="PulseJobType Diagram" /></td>
</tr>
<tr>
<td>type</td>
<td>restriction of <code>xs:string</code></td>
</tr>
</tbody>
</table>
| properties | isRef 0  
              minOcc 0  
              maxOcc 1  
              content `simple`  
              default `True` |
| facets | enumeration True  
|        | enumeration False |
| source | `<xs:element name="PulseJobType" default="True" minOccurs="0">  
|        | `<xs:simpleType>  
|        | `<xs:restriction base="xs:string">  
|        | `<xs:enumeration value="True"/`>  
|        | `<xs:enumeration value="False"/>  
|        | `<xs:restriction>  
|        | </xs:simpleType>  
|        | </xs:element> |

element JobOutputOptionsType/JobType/RectanJobType

diagram

![Diagram](image)

properties

- isRef 0
- minOccurs 0
- maxOccurs 1
- content complex

children PositionConfiguration RobotPosture

source `<xs:element name="RectanJobType" minOccurs="0">  
|        | `<xs:complexType>  
|        | `<xs:sequence>  
|        | `<xs:element name="PositionConfiguration" type="PositionConfigurationType"/>  
|        | `<xs:element name="RobotPosture">  
|        | `<xs:complexType>  
|        | `<xs:all>  
|        | `<xs:element name="FlipPosture" default="0">  
|        | `<xs:simpleType>  
|        | `<xs:restriction base="xs:integer">  
|        | `<xs:enumeration value="0"/>  
|        | `<xs:enumeration value="1"/>  
|        | </xs:restriction>  
|        | </xs:simpleType>  
|        | </xs:element>  
|        | `<xs:element name="UpperPosture" default="0">  
|        | `<xs:simpleType>  
|        | `<xs:restriction base="xs:integer">  
|        | `<xs:enumeration value="0"/>  
|        | `<xs:enumeration value="1"/>  
|        | </xs:restriction>  
|        | </xs:simpleType>  
|        | </xs:element>  
|        | `<xs:element name="FrontPosture" default="0">  
|        | `<xs:simpleType>  
|        | `<xs:restriction base="xs:integer">  
|        | `<xs:enumeration value="0"/>  
|        | `<xs:enumeration value="1"/>  
|        | </xs:restriction>  
|        | </xs:simpleType>  
|        | </xs:all>  
|        | </xs:complexType>  
|        | </xs:element>`
element JobOutputOptionsType/JobType/RectanJobType/PositionConfiguration

diagram

PositionConfiguration

RobotFrame

BaseFrame

UserFrame

properties

isRef 0
content complex

children

RobotFrame BaseFrame UserFrame

source

<xs:element name="PositionConfiguration" type="PositionConfigurationType"/>

element JobOutputOptionsType/JobType/RectanJobType/RobotPosture
source

```xml
<xs:element name="RobotPosture">
  <xs:complexType>
    <xs:all>
      <xs:element name="FlipPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="UpperPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="FrontPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="RPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="TPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:all>
  </xs:complexType>
</xs:element>
```
element `JobOutputOptionsType/JobType/RectanJobType/RobotPosture/FlipPosture`

<table>
<thead>
<tr>
<th>diagram</th>
<th>FlipPosture</th>
</tr>
</thead>
</table>

| type     | restriction of `xs:integer` |
| properties | isRef 0  
             | content simple  
             | default 0  |
| facets   | enumeration 0  
             | enumeration 1  |

| source | `<xs:element name="FlipPosture" default="0">`  
        | `<xs:restriction base="xs:integer">`  
        | `<xs:enumeration value="0"/>`  
        | `<xs:enumeration value="1"/>`  
        | </xs:restriction>  
        | </xs:element>` |

element `JobOutputOptionsType/JobType/RectanJobType/RobotPosture/UpperPosture`

<table>
<thead>
<tr>
<th>diagram</th>
<th>UpperPosture</th>
</tr>
</thead>
</table>

| type     | restriction of `xs:integer` |
| properties | isRef 0  
             | content simple  
             | default 0  |
| facets   | enumeration 0  
             | enumeration 1  |

| source | `<xs:element name="UpperPosture" default="0">`  
        | `<xs:restriction base="xs:integer">`  
        | `<xs:enumeration value="0"/>`  
        | `<xs:enumeration value="1"/>`  
        | </xs:restriction>  
        | </xs:element>` |

element `JobOutputOptionsType/JobType/RectanJobType/RobotPosture/FrontPosture`
### FrontPosture

**Diagram**: FrontPosture

**Type**: Restriction of xs:integer

**Properties**

- `isRef`: 0
- `content`: Simple
- `default`: 0

**Facets**

- Enumeration: 0
- Enumeration: 1

**Source**

```xml
<xs:element name="FrontPosture" default="0">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:enumeration value="0"/>
      <xs:enumeration value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

### RPosture

**Diagram**: RPosture

**Type**: Restriction of xs:integer

**Properties**

- `isRef`: 0
- `content`: Simple
- `default`: 0

**Facets**

- Enumeration: 0
- Enumeration: 1

**Source**

```xml
<xs:element name="RPosture" default="0">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:enumeration value="0"/>
      <xs:enumeration value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

### TPosture

**Diagram**: TPosture

**Type**: Restriction of xs:integer

**Properties**

- `isRef`: 0
- `content`: Simple
- `default`: 0

**Facets**

- Enumeration: 0
- Enumeration: 1

**Source**

```xml
<xs:element name="TPosture" default="0">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:enumeration value="0"/>
      <xs:enumeration value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```
This command is used to list all of the G Code functions that will cause robot motion. It is used by the G Code Converter during the G Code to Inform Conversion to parse all commands with multiple instructions and place the motion functions last, next to the XYZ position values.
Like ThrowAwayLines but allows the user to have multiple lines types of comment lines. i.e. lines that are ignored in the translation process.

```xml
<xs:complexType name="ParseGroupsThrowAwayLinesType">
  <xs:annotation>
    <xs:documentation>Like ThrowAwayLines but allows the user to have multiple lines types of comment lines. i.e. lines that are ignored in the translation process.</xs:documentation>
  </xs:annotation>
  <xs:attribute name="G1" type="xs:string" use="required"/>
  <xs:attribute name="G2" type="xs:string" use="optional"/>
  <xs:attribute name="G3" type="xs:string" use="optional"/>
</xs:complexType>
```
used by element `FileConfigurationType/ParseGroups`

<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>G1</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G2</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G3</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G4</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G5</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G6</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G7</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G8</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G9</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G10</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G11</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G12</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G13</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G14</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G15</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G16</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G17</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
<xs:complexType name="ParseGroupsType">
    <xs:attribute name="G1" type="xs:string" use="optional"/>
    <xs:attribute name="G2" type="xs:string" use="optional"/>
    <xs:attribute name="G3" type="xs:string" use="optional"/>
    <xs:attribute name="G4" type="xs:string" use="optional"/>
    <xs:attribute name="G5" type="xs:string" use="optional"/>
    <xs:attribute name="G6" type="xs:string" use="optional"/>
    <xs:attribute name="G7" type="xs:string" use="optional"/>
    <xs:attribute name="G8" type="xs:string" use="optional"/>
    <xs:attribute name="G9" type="xs:string" use="optional"/>
    <xs:attribute name="G10" type="xs:string" use="optional"/>
    <xs:attribute name="G11" type="xs:string" use="optional"/>
    <xs:attribute name="G12" type="xs:string" use="optional"/>
    <xs:attribute name="G13" type="xs:string" use="optional"/>
    <xs:attribute name="G14" type="xs:string" use="optional"/>
    <xs:attribute name="G15" type="xs:string" use="optional"/>
    <xs:attribute name="G16" type="xs:string" use="optional"/>
    <xs:attribute name="G17" type="xs:string" use="optional"/>
</xs:complexType>

attribute ParseGroupsType/@G1
  type xs:string
  properties isRef 0
         use optional
  source <xs:attribute name="G1" type="xs:string" use="optional"/>

attribute ParseGroupsType/@G2
  type xs:string
  properties isRef 0
         use optional
  source <xs:attribute name="G2" type="xs:string" use="optional"/>

attribute ParseGroupsType/@G3
  type xs:string
  properties isRef 0
         use optional
  source <xs:attribute name="G3" type="xs:string" use="optional"/>

attribute ParseGroupsType/@G4
  type xs:string
  properties isRef 0
         use optional
  source <xs:attribute name="G4" type="xs:string" use="optional"/>

attribute ParseGroupsType/@G5
<table>
<thead>
<tr>
<th>attribute ParseGroupsType/@G5</th>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
</table>
|                              | properties | isRef 0  
|                               |            | use optional |
|                              | source | <xs:attribute name="G5" type="xs:string" use="optional"/> |

<table>
<thead>
<tr>
<th>attribute ParseGroupsType/@G6</th>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
</table>
|                              | properties | isRef 0  
|                               |            | use optional |
|                              | source | <xs:attribute name="G6" type="xs:string" use="optional"/> |

<table>
<thead>
<tr>
<th>attribute ParseGroupsType/@G7</th>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
</table>
|                              | properties | isRef 0  
|                               |            | use optional |
|                              | source | <xs:attribute name="G7" type="xs:string" use="optional"/> |

<table>
<thead>
<tr>
<th>attribute ParseGroupsType/@G8</th>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
</table>
|                              | properties | isRef 0  
|                               |            | use optional |
|                              | source | <xs:attribute name="G8" type="xs:string" use="optional"/> |

<table>
<thead>
<tr>
<th>attribute ParseGroupsType/@G9</th>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
</table>
|                              | properties | isRef 0  
|                               |            | use optional |
|                              | source | <xs:attribute name="G9" type="xs:string" use="optional"/> |

<table>
<thead>
<tr>
<th>attribute ParseGroupsType/@G10</th>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
</table>
|                              | properties | isRef 0  
|                               |            | use optional |
|                              | source | <xs:attribute name="G10" type="xs:string" use="optional"/> |

<table>
<thead>
<tr>
<th>attribute ParseGroupsType/@G11</th>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
</table>
|                              | properties | isRef 0  
<p>|                               |            | use optional |
|                              | source | &lt;xs:attribute name=&quot;G11&quot; type=&quot;xs:string&quot; use=&quot;optional&quot;/&gt; |</p>
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Properties</th>
<th>Use</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParseGroupsType/@G12</td>
<td>xs:string</td>
<td>isRef 0</td>
<td>optional</td>
<td>&lt;xs:attribute name=&quot;G12&quot; type=&quot;xs:string&quot; use=&quot;optional&quot;/&gt;</td>
</tr>
<tr>
<td>ParseGroupsType/@G13</td>
<td>xs:string</td>
<td>isRef 0</td>
<td>optional</td>
<td>&lt;xs:attribute name=&quot;G13&quot; type=&quot;xs:string&quot; use=&quot;optional&quot;/&gt;</td>
</tr>
<tr>
<td>ParseGroupsType/@G14</td>
<td>xs:string</td>
<td>isRef 0</td>
<td>optional</td>
<td>&lt;xs:attribute name=&quot;G14&quot; type=&quot;xs:string&quot; use=&quot;optional&quot;/&gt;</td>
</tr>
<tr>
<td>ParseGroupsType/@G15</td>
<td>xs:string</td>
<td>isRef 0</td>
<td>optional</td>
<td>&lt;xs:attribute name=&quot;G15&quot; type=&quot;xs:string&quot; use=&quot;optional&quot;/&gt;</td>
</tr>
<tr>
<td>ParseGroupsType/@G16</td>
<td>xs:string</td>
<td>isRef 0</td>
<td>optional</td>
<td>&lt;xs:attribute name=&quot;G16&quot; type=&quot;xs:string&quot; use=&quot;optional&quot;/&gt;</td>
</tr>
<tr>
<td>ParseGroupsType/@G17</td>
<td>xs:string</td>
<td>isRef 0</td>
<td>optional</td>
<td>&lt;xs:attribute name=&quot;G17&quot; type=&quot;xs:string&quot; use=&quot;optional&quot;/&gt;</td>
</tr>
</tbody>
</table>

**complexType** PositionConfigurationType
**PositionConfigurationType**

- **RobotFrame**
  - Type: `xs:string`
  - Default: `True`
  - MinOcc: `0`
  - MaxOcc: `1`
  - Content: `simple`
  - Source: `xs:element name="RobotFrame" type="xs:string" default="True" minOccurs="0"`;

- **BaseFrame**
  - Type: `xs:string`
  - Default: `False`
  - MinOcc: `0`
  - MaxOcc: `1`
  - Content: `simple`
  - Source: `xs:element name="BaseFrame" default="False" minOccurs="0"`;

- **UserFrame**
  - Type: `xs:integer`
  - Total: `2`
  - Fraction: `0`
  - Min: `0`
  - Max: `25`
  - Pattern: `\d{1,2}`
  - Source: `xs:element name="UserFrame" default="1" minOccurs="0"`;

**used by**
- `InputPositionDataType/InputPositionDataType/Rectan/PositionConfiguration`
- `JobOutputOptionsType/JobType/RectanJobType/PositionConfiguration`
### diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:string</th>
</tr>
</thead>
</table>
| properties | isRef 0  
| | minOccurs 0  
| | maxOcc 1  
| | content simple  
| | default False  |
| facets | enumeration True  
| | enumeration False  |
| source | `<xs:element name="BaseFrame" default="False" minOccurs="0">  
| | `<xs:simpleType>  
| | `<xs:restriction base="xs:string">  
| | `<xs:enumeration value="True"/>  
| | `<xs:enumeration value="False"/>  
| | `</xs:restriction>  
| | `</xs:simpleType>  
| | `</xs:element>` |}

**element** PositionConfigurationType/UserFrame

<table>
<thead>
<tr>
<th>diagram</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:integer</td>
</tr>
</tbody>
</table>
| properties | isRef 0  
| | minOccurs 0  
| | maxOcc 1  
| | content simple  
| | default 1 |
| facets | minOccurs 0  
| | maxInclusive 25  
| | totalDigits 2  
| | fractionDigits 0  
| | pattern \\d\{1,2\} |
| source | `<xs:element name="UserFrame" default="1" minOccurs="0">  
| | `<xs:simpleType>  
| | `<xs:restriction base="xs:integer">  
| | `<xs:totalDigits value="2"/>  
| | `<xs:fractionDigits value="0"/>  
| | `<xs:minInclusive value="0"/>  
| | `<xs:maxInclusive value="25"/>  
| | `<xs:pattern value="\\d\{1,2\}"/>  
| | `</xs:restriction>  
| | `</xs:simpleType>  
| | `</xs:element>` |}

### complexType PositionConfigurationPulseType
<xs:complexType name="PositionConfigurationPulseType">
    <xs:sequence>
        <xs:element name="BaseFrame" default="False" minOccurs="0">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="True"/>
                    <xs:enumeration value="False"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="RobotFrame" type="xs:string" default="True" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

**element** PositionConfigurationPulseType/BaseFrame

<table>
<thead>
<tr>
<th>Diagram</th>
<th>type</th>
<th>restriction of xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>properties</td>
<td>isRef 0, minOcc 0, maxOcc 1, content simple, default False</td>
</tr>
<tr>
<td></td>
<td>facets</td>
<td>enumeration True, enumeration False</td>
</tr>
</tbody>
</table>
| source | <xs:element name="BaseFrame" default="False" minOccurs="0">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="True"/>
                    <xs:enumeration value="False"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element> |

**element** PositionConfigurationPulseType/RobotFrame

<table>
<thead>
<tr>
<th>Diagram</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>xs:string</td>
</tr>
</tbody>
</table>

**source** element InputPositionDataType/InputPositionDataPulse/PositionConfigurationPulse

**children** BaseFrame, RobotFrame

**used by**
complexType `ReferenceModelType`

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="ReferenceModelType" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>extension of <code>ReferenceTypeBase</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>base <code>ReferenceTypeBase</code></td>
</tr>
<tr>
<td>facets</td>
<td>enumeration UF1</td>
</tr>
<tr>
<td></td>
<td>enumeration UF2</td>
</tr>
<tr>
<td></td>
<td>enumeration UF3</td>
</tr>
<tr>
<td></td>
<td>enumeration UF4</td>
</tr>
<tr>
<td></td>
<td>enumeration UF5</td>
</tr>
<tr>
<td></td>
<td>enumeration UF6</td>
</tr>
<tr>
<td></td>
<td>enumeration UF7</td>
</tr>
<tr>
<td></td>
<td>enumeration UF8</td>
</tr>
<tr>
<td></td>
<td>enumeration UF9</td>
</tr>
<tr>
<td></td>
<td>enumeration UF10</td>
</tr>
<tr>
<td></td>
<td>enumeration UF11</td>
</tr>
<tr>
<td></td>
<td>enumeration UF12</td>
</tr>
<tr>
<td></td>
<td>enumeration UF13</td>
</tr>
<tr>
<td></td>
<td>enumeration UF14</td>
</tr>
<tr>
<td></td>
<td>enumeration UF15</td>
</tr>
<tr>
<td></td>
<td>enumeration UF16</td>
</tr>
<tr>
<td></td>
<td>enumeration UF17</td>
</tr>
<tr>
<td></td>
<td>enumeration UF18</td>
</tr>
<tr>
<td></td>
<td>enumeration UF19</td>
</tr>
<tr>
<td></td>
<td>enumeration UF20</td>
</tr>
<tr>
<td></td>
<td>enumeration UF21</td>
</tr>
<tr>
<td></td>
<td>enumeration UF22</td>
</tr>
<tr>
<td></td>
<td>enumeration UF23</td>
</tr>
<tr>
<td></td>
<td>enumeration UF24</td>
</tr>
<tr>
<td></td>
<td>enumeration UF25</td>
</tr>
</tbody>
</table>

source

```xml
<xs:complexType name="ReferenceModelType">
  <xs:simpleContent>
    <xs:extension base="ReferenceTypeBase"/>
  </xs:simpleContent>
</xs:complexType>
```

complexType `RotatingStationJobDataPositionType`
source
<xs:complexType name="RotatingStationJobDataPositionType">
  <xs:sequence>
    <xs:element name="RotationStationName" default="ST1"/>
    <xs:element name="PositionUnits" default="Degrees" minOccurs="0">
      <xs:restriction base="xs:string">
        <xs:pattern value="Pulse"/>
        <xs:pattern value="Degrees"/>
        <xs:pattern value="Radians"/>
      </xs:restriction>
    </xs:element>
    <xs:element name="PositionValue" default="0.0">
      <xs:restriction base="xs:double">
        <xs:minInclusive value="-10000.0"/>
        <xs:maxInclusive value="10000.0"/>
      </xs:restriction>
    </xs:element>
    <xs:element name="RotatingVelocity" default="25" minOccurs="0">
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="100"/>
      </xs:restriction>
    </xs:element>
    <xs:element name="RotatingMotionType" default="Joint" minOccurs="0">
      <xs:restriction base="xs:string">
        <xs:pattern value="Joint"/>
        <xs:pattern value="Linear"/>
        <xs:pattern value="Spherical"/>
        <xs:pattern value="Polar"/>
        <xs:pattern value="Cylindrical"/>
        <xs:pattern value="Cone"/>
        <xs:pattern value="Conical"/>
        <xs:pattern value="Pendulum"/>
        <xs:pattern value="Screw"/>
        <xs:pattern value="Spiral"/>
        <xs:pattern value="Helicoidal"/>
        <xs:pattern value="Helical"/>
        <xs:pattern value="Lateral"/>
        <xs:pattern value="LateralLinear"/>
        <xs:pattern value="LateralSpherical"/>
        <xs:pattern value="LateralPolar"/>
        <xs:pattern value="LateralCylindrical"/>
        <xs:pattern value="LateralConical"/>
        <xs:pattern value="LateralCone"/>
        <xs:pattern value="LateralScrew"/>
        <xs:pattern value="LateralSpiral"/>
        <xs:pattern value="LateralHelicoidal"/>
        <xs:pattern value="LateralHelical"/>
        <xs:pattern value="LateralLateral"/>
        <xs:pattern value="LateralLinearLateral"/>
        <xs:pattern value="LateralSphericalLateral"/>
        <xs:pattern value="LateralPolarLateral"/>
        <xs:pattern value="LateralCylindricalLateral"/>
        <xs:pattern value="LateralConicalLateral"/>
        <xs:pattern value="LateralConeLateral"/>...
### element **RotatingStationJobDataPositionType/RotationStationName**

- **Diagram:** ![RotationStationName](image)
- **Type:** restriction of `xs:string`
- **Properties:**
  - `isRef`: 0
  - `content`: simple
  - `default`: ST1
- **Facets:**
  - pattern ST1
  - pattern ST2
  - pattern ST3
  - pattern ST4
  - pattern ST5

- **Source**
  ```xml
  <xs:element name="RotationStationName" default="ST1">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:pattern value="ST1"/>
        <xs:pattern value="ST2"/>
        <xs:pattern value="ST3"/>
        <xs:pattern value="ST4"/>
        <xs:pattern value="ST5"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  ```

### element **RotatingStationJobDataPositionType/PositionUnits**

- **Diagram:** ![PositionUnits](image)
- **Type:** restriction of `xs:string`
- **Properties:**
  - `isRef`: 0
  - `minOcc`: 0
  - `maxOcc`: 1
  - `content`: simple
  - `default`: Degrees
- **Facets:**
  - pattern Pulse
  - pattern Degrees
  - pattern Radians

- **Source**
  ```xml
  ```
element RotatingStationJobDataPositionType/PositionValue

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:double</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>content simple</td>
</tr>
<tr>
<td></td>
<td>default 0.0</td>
</tr>
<tr>
<td>facets</td>
<td>minInclusive -10000.0</td>
</tr>
<tr>
<td></td>
<td>maxInclusive 10000.0</td>
</tr>
</tbody>
</table>

source

<xs:element name="PositionValue" default="0.0">
  <xs:simpleType>
    <xs:restriction base="xs:double">
      <xs:minInclusive value="-10000.0"/>
      <xs:maxInclusive value="10000.0"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

element RotatingStationJobDataPositionType/RotatingVelocity

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>minOcc 0</td>
</tr>
<tr>
<td></td>
<td>maxOcc 1</td>
</tr>
<tr>
<td></td>
<td>content simple</td>
</tr>
<tr>
<td></td>
<td>default 25</td>
</tr>
<tr>
<td>facets</td>
<td>minInclusive 1</td>
</tr>
<tr>
<td></td>
<td>maxInclusive 100</td>
</tr>
</tbody>
</table>

source

<xs:element name="RotatingVelocity" default="25" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="100"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
element RotatingStationJobDataPositionType/RotatingMotionType

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="RotatingMotionType diagram" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:string</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>isRef 0</td>
</tr>
<tr>
<td>minOccurs 0</td>
</tr>
<tr>
<td>maxOcc 1</td>
</tr>
<tr>
<td>content simple</td>
</tr>
<tr>
<td>default Joint</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>facets</th>
</tr>
</thead>
<tbody>
<tr>
<td>pattern Joint</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;xs:element name=&quot;RotatingMotionType&quot; default=&quot;Joint&quot; minOccurs=&quot;0&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:simpleType&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:restriction base=&quot;xs:string&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:pattern value=&quot;Joint&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/xs:restriction&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/xs:simpleType&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/xs:element&gt;</code></td>
</tr>
</tbody>
</table>

complexType RotatingStationSetupType

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="RotatingStationSetupType diagram" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>children</th>
</tr>
</thead>
<tbody>
<tr>
<td>RotationStationName RotatingStationUserFrameNumber PartCordinatesRotateWithStation DefaultRotatingVelocity DefaultRotatingMotionType DefaultRotatingUnits RotatingStationToCNCAngleMapping</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>used by</th>
</tr>
</thead>
<tbody>
<tr>
<td>element FileConfigurationType/RotatingStationSetup</td>
</tr>
</tbody>
</table>
<xs:complexType name="RotatingStationSetupType">
  <xs:sequence>
    <xs:element name="RotationStationName" default="ST1">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="ST1"/>
          <xs:enumeration value="ST2"/>
          <xs:enumeration value="ST3"/>
          <xs:enumeration value="ST4"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="RotatingStationUserFrameNumber" default="1">
      <xs:simpleType>
        <xs:restriction base="xs:integer">
          <xs:totalDigits value="2"/>
          <xs:fractionDigits value="0"/>
          <xs:minInclusive value="1"/>
          <xs:maxInclusive value="24"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="PartCordinatesRotateWithStation" default="True">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
          <xs:enumeration value="False"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="DefaultRotatingVelocity" default="25">
      <xs:simpleType>
        <xs:restriction base="xs:integer">
          <xs:minInclusive value="0"/>
          <xs:maxInclusive value="100"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="DefaultRotatingMotionType" default="Joint">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:pattern value="Joint"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="DefaultRotatingUnits" default="Degrees">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:pattern value="EncoderCounts"/>
          <xs:pattern value="Degrees"/>
          <xs:pattern value="Radians"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
**element RotatingStationSetupType/RotationStationName**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="Diagram" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:string</td>
</tr>
</tbody>
</table>
| properties | isRef 0  
            content simple  
            default ST1 |
| facets  | enumeration ST1  
            enumeration ST2  
            enumeration ST3  
            enumeration ST4 |
| source  | `<xs:element name="RotationStationName" default="ST1">  
            <xs:simpleType>  
            <xs:restriction base="xs:string">  
            <xs:enumeration value="ST1"/>  
            <xs:enumeration value="ST2"/>  
            <xs:enumeration value="ST3"/>  
            <xs:enumeration value="ST4"/>  
            </xs:restriction>  
            </xs:simpleType>  
            </xs:element>` |

**element RotatingStationSetupType/RotatingStationUserFrameNumber**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="Diagram" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:integer</td>
</tr>
</tbody>
</table>
| properties | isRef 0  
            content simple  
            default 1 |
| facets  | minInclusive 1  
            maxInclusive 24  
            totalDigits 2  
            fractionDigits 0 |
| source  | `<xs:element name="RotatingStationUserFrameNumber" default="1">  
            <xs:simpleType>  
            <xs:restriction base="xs:integer">  
            <xs:totalDigits value="2"/>  
            <xs:fractionDigits value="0"/>  
            <xs:minInclusive value="1"/>  
            <xs:maxInclusive value="24"/>  
            </xs:restriction>  
            </xs:simpleType>  
            </xs:element>` |
### RotatingStationType/PartCordinatesRotateWithStation

<table>
<thead>
<tr>
<th>Diagram</th>
<th>PartCordinatesRotateWithStation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>restriction of xs:string</td>
</tr>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>IsRef</td>
<td>0</td>
</tr>
<tr>
<td>Content</td>
<td>simple</td>
</tr>
<tr>
<td>Default</td>
<td>True</td>
</tr>
<tr>
<td>Facets</td>
<td></td>
</tr>
<tr>
<td>Enumeration</td>
<td>True</td>
</tr>
<tr>
<td>Enumeration</td>
<td>False</td>
</tr>
</tbody>
</table>

Source:

```xml
<xs:element name="PartCordinatesRotateWithStation" default="True">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="True"/>
      <xs:enumeration value="False"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

### RotatingStationType/DefaultRotatingVelocity

<table>
<thead>
<tr>
<th>Diagram</th>
<th>DefaultRotatingVelocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>restriction of xs:integer</td>
</tr>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>IsRef</td>
<td>0</td>
</tr>
<tr>
<td>Content</td>
<td>simple</td>
</tr>
<tr>
<td>Default</td>
<td>25</td>
</tr>
<tr>
<td>Facets</td>
<td></td>
</tr>
<tr>
<td>MinInclusive</td>
<td>0</td>
</tr>
<tr>
<td>MaxInclusive</td>
<td>100</td>
</tr>
</tbody>
</table>

Source:

```xml
<xs:element name="DefaultRotatingVelocity" default="25">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="100"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

### RotatingStationType/DefaultRotatingMotionType

<table>
<thead>
<tr>
<th>Diagram</th>
<th>DefaultRotatingMotionType</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>restriction of xs:string</td>
</tr>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>IsRef</td>
<td>0</td>
</tr>
<tr>
<td>Content</td>
<td>simple</td>
</tr>
<tr>
<td>Default</td>
<td>Joint</td>
</tr>
<tr>
<td>Facets</td>
<td></td>
</tr>
<tr>
<td>Pattern</td>
<td>Joint</td>
</tr>
</tbody>
</table>
element RotatingStationSetupType/DefaultRotatingUnits

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:string</th>
</tr>
</thead>
</table>
| properties | isRef 0  
              content simple  
              default Degrees |
| facets  | pattern EncoderCounts  
              pattern Degrees  
              pattern Radians |

source

<xs:element name="DefaultRotatingUnits" default="Degrees">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="EncoderCounts"/>
      <xs:pattern value="Degrees"/>
      <xs:pattern value="Radians"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

element RotatingStationSetupType/RotatingStationToCNCAngleMapping

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>RotatingStationToCNCAngleMappingType</th>
</tr>
</thead>
</table>
| properties    | isRef 0  
              content complex |
| children      | StationtoCNCAxis |

source

<xs:element name="RotatingStationToCNCAngleMapping" type="RotatingStationToCNCAngleMappingType"/>
element RotatingStationToCNCAngleMappingType/StationtoCNCAxis
<xs:element name="StationtoCNCAxis" maxOccurs="3">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="StationAxisNumber" default="1">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="1"/>
            <xs:enumeration value="2"/>
            <xs:enumeration value="3"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="CNCAxisName" default="A">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="A"/>
            <xs:enumeration value="B"/>
            <xs:enumeration value="C"/>
            <xs:enumeration value="D"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="NegageCNCAxis" default="False">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="False"/>
            <xs:enumeration value="True"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<table>
<thead>
<tr>
<th>properties</th>
<th>isRef</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>content</td>
<td>simple</td>
<td></td>
</tr>
<tr>
<td>default</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>facets</th>
<th>enumeration</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>enumeration</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>enumeration</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>source</th>
<th>&lt;xs:element name=&quot;StationAxisNumber&quot; default=&quot;1&quot;&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><a href="">xs:simpleType</a></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:restriction base=&quot;xs:integer&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;1&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;2&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;3&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:restriction&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:simpleType&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:element&gt;</td>
</tr>
</tbody>
</table>

**element RotatingStationToCNCAngleMappingType/StationtoCNCAxis/CNCAxisName**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="CNCAxisName" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:string</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>properties</th>
<th>isRef</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>content</td>
<td>simple</td>
<td></td>
</tr>
<tr>
<td>default</td>
<td>A</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>facets</th>
<th>enumeration</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>enumeration</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>enumeration</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>enumeration</td>
<td>D</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>source</th>
<th>&lt;xs:element name=&quot;CNCAxisName&quot; default=&quot;A&quot;&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><a href="">xs:simpleType</a></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:restriction base=&quot;xs:string&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;A&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;B&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;C&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;D&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:restriction&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:simpleType&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:element&gt;</td>
</tr>
</tbody>
</table>

**element RotatingStationToCNCAngleMappingType/StationtoCNCAxis/NegageCNCAxis**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="NegageCNCAxis" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:string</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>properties</th>
<th>isRef</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>content</td>
<td>simple</td>
<td></td>
</tr>
<tr>
<td>default</td>
<td>False</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>facets</th>
<th>enumeration</th>
<th>False</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>enumeration</td>
<td>True</td>
</tr>
</tbody>
</table>


ComplexType `Search`

```xml
<xs:complexType name="Search">
  <xs:sequence>
    <xs:element ref="DistVar"/>
    <xs:element name="State" default="Off">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="On"/>
          <xs:enumeration value="Off"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Input" default="1">
      <xs:simpleType>
        <xs:restriction base="xs:integer">
          <xs:totalDigits value="3"/>
          <xs:fractionDigits value="0"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Const" type="xs:short" default="0"/>
  </xs:sequence>
</xs:complexType>
```

Element `Search/State`

```xml
<xs:element name="State" default="Off">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="On"/>
      <xs:enumeration value="Off"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Type restriction of `xs:string`
element Search/Input

source

<xs:element name="State" default="Off">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="On"/>
      <xs:enumeration value="Off"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

element Search/Input

source

<xs:element name="Input" default="1">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:totalDigits value="3"/>
      <xs:fractionDigits value="0"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

element Search/Const

source

<xs:element name="Const" type="xs:short" default="0"/>

complexType ShiftOffset
### ShiftOffset

This element and its sub elements load a PVariable element from a D Variable and then.

#### Children
- ShiftOffsetDVarIndex
- ShiftFunctionType
- ShiftSourceNum
- ShiftReferenceModel

#### Used by
- Element: ShiftOffset

#### Annotation Documentation
- **This element and its sub elements load a PVariable element from a D Variable and then**

#### Source
```xml
<xs:complexType name="ShiftOffset">
  <xs:annotation>
    <xs:documentation>This element and its sub elements load a PVariable element from a D Variable and then</xs:documentation>
  </xs:annotation>
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element ref="ShiftOffsetDVarIndex" minOccurs="0"/>
    <xs:element ref="ShiftFunctionType" minOccurs="0"/>
    <xs:element ref="ShiftSourceNum" minOccurs="0"/>
    <xs:element ref="ShiftReferenceModel" minOccurs="0"/>
  </xs:choice>
</xs:complexType>
```

### ThreeDOFType

Supplies the default orientation of the Robot Tool. This is the value that makes the last length of tool perpendicular to the part work plane.

#### Diagram

[Diagram of ThreeDOFType with attributes Rx, Ry, Rz]
used by elements ThreeDOF FileConfigurationType/ThreeDOF

attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rx</td>
<td>xs:decimal</td>
<td>required</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ry</td>
<td>xs:decimal</td>
<td>required</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rz</td>
<td>derived by: xs:decimal</td>
<td>required</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

annotation documentation
Supplies the default orientation of the Robot Tool. This is the value that makes the last length of tool perpendicular to the part work plane.

source

```xml
<xs:complexType name="ThreeDOFType">
  <xs:annotation>
    <xs:documentation>Supplies the default orientation of the Robot Tool. This is the value that makes the last length of tool perpendicular to the part work plane.</xs:documentation>
  </xs:annotation>
  <xs:attribute name="Rx" type="xs:decimal" use="required"/>
  <xs:attribute name="Ry" type="xs:decimal" use="required"/>
  <xs:attribute name="Rz" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:decimal"/>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
```

attribute ThreeDOFType/@Rx

type xs:decimal

properties isRef 0
use required

source

```xml
<xs:attribute name="Rx" type="xs:decimal" use="required"/>
```

attribute ThreeDOFType/@Ry

type xs:decimal

properties isRef 0
use required

source

```xml
<xs:attribute name="Ry" type="xs:decimal" use="required"/>
```

attribute ThreeDOFType/@Rz

type restriction of xs:decimal

properties isRef 0
use required

source

```xml
<xs:attribute name="Rz" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:decimal"/>
  </xs:simpleType>
</xs:attribute>
```
**complexType UnitsType**

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="UnitsType Diagram" /></td>
</tr>
</tbody>
</table>

children: `LengthUnits`, `AngleUnits`, `TimeUnits`

used by: `FileConfigurationType/Units`

source:
```xml
<xs:complexType name="UnitsType">
  <xs:sequence>
    <xs:element ref="LengthUnits"/>
    <xs:element ref="AngleUnits"/>
    <xs:element ref="TimeUnits"/>
  </xs:sequence>
</xs:complexType>
```

**simpleType ReferenceTypeBase**

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of <code>xs:string</code></th>
</tr>
</thead>
</table>

used by: `complexType ReferenceModelType`

facets:
- `enumeration UF1`
- `enumeration UF2`
- `enumeration UF3`
- `enumeration UF4`
- `enumeration UF5`
- `enumeration UF6`
- `enumeration UF7`
- `enumeration UF8`
- `enumeration UF9`
- `enumeration UF10`
- `enumeration UF11`
- `enumeration UF12`
- `enumeration UF13`
- `enumeration UF14`
- `enumeration UF15`
- `enumeration UF16`
- `enumeration UF17`
- `enumeration UF18`
- `enumeration UF19`
- `enumeration UF20`
- `enumeration UF21`
- `enumeration UF22`
- `enumeration UF23`
- `enumeration UF24`
- `enumeration UF25`
<xs:simpleType name="ReferenceTypeBase">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UF1"/>
    <xs:enumeration value="UF2"/>
    <xs:enumeration value="UF3"/>
    <xs:enumeration value="UF4"/>
    <xs:enumeration value="UF5"/>
    <xs:enumeration value="UF6"/>
    <xs:enumeration value="UF7"/>
    <xs:enumeration value="UF8"/>
    <xs:enumeration value="UF9"/>
    <xs:enumeration value="UF10"/>
    <xs:enumeration value="UF11"/>
    <xs:enumeration value="UF12"/>
    <xs:enumeration value="UF13"/>
    <xs:enumeration value="UF14"/>
    <xs:enumeration value="UF15"/>
    <xs:enumeration value="UF16"/>
    <xs:enumeration value="UF17"/>
    <xs:enumeration value="UF18"/>
    <xs:enumeration value="UF19"/>
    <xs:enumeration value="UF20"/>
    <xs:enumeration value="UF21"/>
    <xs:enumeration value="UF22"/>
    <xs:enumeration value="UF23"/>
    <xs:enumeration value="UF24"/>
    <xs:enumeration value="UF25"/>
  </xs:restriction>
</xs:simpleType>
element AngleSystem

- **Diagram**: This element specifies the reference system for position orientation angles. Values are YPR and Euler. YPR is the standard MotoSimEG angle system. Euler means that the angle system is specified in terms of the Euler Angle system. This is a system that is used in the aircraft industry.

- **Type**: restriction of `xs:string`

- **Properties**: content simple

- **Used by**: element `Root/Settings`

- **Facets**: enumeration YPR, enumeration Euler

- **Annotation**: documentation

This element specifies the reference system for position orientation angles. Values are YPR and Euler. YPR is the standard MotoSimEG angle system. Euler means that the angle system is specified in terms of the Euler Angle system. This is a system that is used in the aircraft industry.
**Source**

```xml
<xs:element name="AngleSystem">
  <xs:annotation>
    <xs:documentation>This element specifics the reference system for position orientation angles. Values are YPR and Euler. YPR is the standard MotoSimEG angle system. Euler means that the angle system is specified in terms of the Euler Angle system. This is a system that is used in the aircraft industry.</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="YPR"/>
      <xs:enumeration value="Euler"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**Element: AngleUnits**

- **Diagram**: ![Diagram of AngleUnits]
- **Type**: restriction of `xs:string`
- **Properties**: content simple
- **Used By**: element `Units`
- **Facets**: enumeration Degrees, enumeration Radians

```xml
<xs:element name="AngleUnits">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Degrees"/>
      <xs:enumeration value="Radians"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**Element: DefaultMotionType**

- **Diagram**: ![Diagram of DefaultMotionType]
- **Type**: restriction of `xs:string`
- **Properties**: content simple

```xml
<xs:element name="DefaultMotionType">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:documentation>When a position command does not include a MotionType command, the DefaultMotionType is used.</xs:documentation>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```
### DefaultMotionType

When a position command does not include a MotionType command, the DefaultMotionType is used.

```xml
<xs:element name="DefaultMotionType">
  <xs:annotation>
    <xs:documentation>When a position command does not include a MotionType command, the DefaultMotionType is used.</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Linear"/>
      <xs:enumeration value="Circular"/>
      <xs:enumeration value="ExternalRef"/>
      <xs:enumeration value="Spline"/>
      <xs:enumeration value="Inc"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

### DefaultVelocity

Default Velocity is the velocity that is used when a position command does not include a velocity command.
Default Velocity is the velocity that is used when a position command does not include a velocity command.

```xml
<xs:element name="DefaultVelocity">
  <xs:annotation>
    <xs:documentation>Default Velocity is the velocity that is used when a position command does not include a velocity command.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Linear" minOccurs="0"/>
      <xs:element ref="Joint" minOccurs="0"/>
      <xs:element name="ExternalRef" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:decimal">
            <xs:fractionDigits value="10"/>
            <xs:totalDigits value="16"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Circular" type="xs:anySimpleType" minOccurs="0"/>
      <xs:element name="Spline" type="xs:anySimpleType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**Element `DefaultVelocity/ExternalRef`**

<table>
<thead>
<tr>
<th>Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="ExternalRef" /></td>
</tr>
</tbody>
</table>

- **Type**: restriction of `xs:decimal`
- **Properties**
  - `isRef`: 0
  - `minOcc`: 0
  - `maxOcc`: 1
  - `content`: `simple`
- **Facets**
  - `totalDigits`: 16
  - `fractionDigits`: 10

**Source**

```xml
<xs:element name="ExternalRef" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:decimal">
      <xs:fractionDigits value="10"/>
      <xs:totalDigits value="16"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**Element `DefaultVelocity/Circular`**

```xml
<xs:element name="Circular" type="xs:anySimpleType" minOccurs="0"/>
<xs:element name="Spline" type="xs:anySimpleType" minOccurs="0"/>
```
### Diagram for `Circular`

| properties | isRef: 0  
|            | minOcc: 0  
|            | maxOcc: 1  
| content    | simple  

```xml
<xs:element name="Circular" type="xs:anySimpleType" minOccurs="0"/>
```

### Diagram for `Spline`

| properties | isRef: 0  
|            | minOcc: 0  
|            | maxOcc: 1  
| content    | simple  

```xml
<xs:element name="Spline" type="xs:anySimpleType" minOccurs="0"/>
```

### Diagram for `Joint`

| type        | restriction of `xs:integer`  
|             | content: simple  
| default     | 20  

```xml
<xs:element name="Joint" default="20">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:totalDigits value="3"/>
      <xs:pattern value="\d{1,3}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```
### element LengthUnits

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="LengthUnits" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of <code>xs:string</code></td>
</tr>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>element <strong>Units</strong></td>
</tr>
</tbody>
</table>
| facets | enumeration **Inches**  
enumeration **Millimeters** |
| source  | `<xs:element name="LengthUnits">`  
`<xs:simpleType>`  
`<xs:restriction base="xs:string">`  
`<xs:enumeration value="Inches"/>`  
`<xs:enumeration value="Millimeters"/>`  
`</xs:restriction>`  
`</xs:simpleType>`  
`</xs:element>` |

### element Linear

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="Linear" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of <code>xs:decimal</code></td>
</tr>
</tbody>
</table>
| properties | content simple  
default 100.0 |
| used by | element **DefaultVelocity** |
| facets | totalDigits 16  
fractionDigits 10  
pattern `(\d{1,6})\.*\d{0,10}` |
| source  | `<xs:element name="Linear" default="100.0">`  
`<xs:simpleType>`  
`<xs:restriction base="xs:decimal">`  
`<xs:fractionDigits value="10"/>`  
`<xs:totalDigits value="16"/>`  
`<xs:pattern value="(\d{1,6})\.*\d{0,10}"/>`  
`</xs:restriction>`  
`</xs:simpleType>`  
`</xs:element>` |

### element ReferenceModel
### referenceModel

**Type:**
restriction of `xs:string`

**Properties:**
- content: simple
- default: UF1

**Used by:**
element `Root/Settings`

**Facets:**
pattern `.{1,15}`

**Annotation Documentation:**
Specifies the user frame number or MotoSimEG Cell Model that is used to specify the origin of the part coordinate system.

**Source:**
```xml
<xs:element name="ReferenceModel" default="UF1">
  <xs:annotation>
    <xs:documentation>Specifies the user frame number or MotoSimEG Cell Model that is used to specify the origin of the part coordinate system. </xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value=".{1,15}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

---

### RobotPosture

**Diagram:**
![RobotPosture Diagram](image)

**Properties:**
- content: complex

**Children:**
- FlipPosture
- UpperPosture
- FrontPosture
- RPosture
- TPosture
<xs:element name="RobotPosture">
  <xs:complexType>
    <xs:all>
      <xs:element name="FlipPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="UpperPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="FrontPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="RPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="TPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:all>
  </xs:complexType>
</xs:element>
element **RobotPosture/FlipPosture**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="FlipPosture" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:integer</td>
</tr>
</tbody>
</table>
| properties | isRef 0  
content simple  
default 0 |
| facets  | enumeration 0  
enumeration 1 |
| source  | `<xs:element name="FlipPosture" default="0">  
  `<xs:simpleType>  
   `<xs:restriction base="xs:integer">  
    `<xs:enumeration value="0"/>  
    `<xs:enumeration value="1"/>  
  </xs:restriction>  
  </xs:simpleType>  
</xs:element>` |

element **RobotPosture/UpperPosture**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="UpperPosture" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:integer</td>
</tr>
</tbody>
</table>
| properties | isRef 0  
content simple  
default 0 |
| facets  | enumeration 0  
enumeration 1 |
| source  | `<xs:element name="UpperPosture" default="0">  
  `<xs:simpleType>  
   `<xs:restriction base="xs:integer">  
    `<xs:enumeration value="0"/>  
    `<xs:enumeration value="1"/>  
  </xs:restriction>  
  </xs:simpleType>  
</xs:element>` |

element **RobotPosture/FrontPosture**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="FrontPosture" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:integer</td>
</tr>
</tbody>
</table>
### element RobotPosture/RPosture

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="RPosture" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:integer</td>
</tr>
<tr>
<td>properties</td>
<td>isRef 0 content simple default 0</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration 0 enumeration 1</td>
</tr>
</tbody>
</table>
| source      | `<xs:element name="RPosture" default="0">  
  `<xs:simpleType>  
   `<xs:restriction base="xs:integer">  
     `<xs:enumeration value="0"/>  
     `<xs:enumeration value="1"/>  
   `/xs:restriction>  
  `/xs:simpleType>  
`/xs:element>` |

### element RobotPosture/TPosture

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="TPosture" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:integer</td>
</tr>
<tr>
<td>properties</td>
<td>isRef 0 content simple default 0</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration 0 enumeration 1</td>
</tr>
</tbody>
</table>
| source      | `<xs:element name="TPosture" default="0">  
  `<xs:simpleType>  
   `<xs:restriction base="xs:integer">  
     `<xs:enumeration value="0"/>  
     `<xs:enumeration value="1"/>  
   `/xs:restriction>  
  `/xs:simpleType>  
`/xs:element>` |
**element Root**

**diagram**

![Diagram of Root element structure](image)

**properties**
- content: complex

**children**
- Settings

**annotation documentation**
Comment describing your root element

**source**

```xml
<xsl:element name="Root">
  <xs:annotation>
    <xs:documentation>Comment describing your root element</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Settings">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="InformFileFormats" type="InformFileFormatsType"/>
            <xs:element ref="AngleSystem"/>
            <xs:element ref="Units"/>
            <xs:element ref="DefaultVelocity"/>
            <xs:element ref="ThreeDOF" minOccurs="0"/>
            <xs:element ref="DefaultMotionType"/>
            <xs:element ref="ReferenceModel"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="ToolNumber" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:int">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
            <xs:enumeration value="2"/>
            <xs:enumeration value="3"/>
            <xs:enumeration value="4"/>
            <xs:enumeration value="5"/>
            <xs:enumeration value="6"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```
element Root/Settings

```
<xs:element name="JobOutputOptions" type="JobOutputOptionsType" minOccurs="0"/>
<xs:element name="ModelOffset" type="ModelOffsetType" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
```
This element, and its sub elements specify the Length, Angle, and Time Units used in this XML File.

**DefaultVelocity**

Default Velocity is the velocity that is used when a position command does not include a velocity command.

**ThreeDOF**

Use this values to set the last length of the tool to be perpendicular to the part coordinate system. Generally this will be 180, 0, X. Where X will be -90, 0, 90, 180, 270 degrees.

**DefaultMotionType**

When a position command does not include a MotionType command, the DefaultMotionType is used.

**ReferenceModel**

Specifies the user frame number or MotoSimEG Cell Model that is used to specify the origin of the part coordinate system.

**ToolNumber**

**JobOutputOptions**

**ModelOffset**

---

<table>
<thead>
<tr>
<th>properties</th>
<th>isRef 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>content</td>
<td>complex</td>
</tr>
</tbody>
</table>

| children    | InformFileFormats AngleSystem Units DefaultVelocity ThreeDOF DefaultMotionType ReferenceModel ToolNumber JobOutputOptions ModelOffset |
<xs:element name="Settings">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="InformFileFormats" type="InformFileFormatsType"/>
      <xs:element ref="AngleSystem"/>
      <xs:element ref="Units"/>
      <xs:element ref="DefaultVelocity"/>
      <xs:element ref="ThreeDOF" minOccurs="0"/>
      <xs:element ref="DefaultMotionType"/>
      <xs:element ref="ReferenceModel"/>
      <xs:element name="ToolNumber" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:int">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
            <xs:enumeration value="2"/>
            <xs:enumeration value="3"/>
            <xs:enumeration value="4"/>
            <xs:enumeration value="5"/>
            <xs:enumeration value="6"/>
            <xs:enumeration value="7"/>
            <xs:enumeration value="8"/>
            <xs:enumeration value="9"/>
            <xs:enumeration value="10"/>
            <xs:enumeration value="11"/>
            <xs:enumeration value="12"/>
            <xs:enumeration value="13"/>
            <xs:enumeration value="14"/>
            <xs:enumeration value="15"/>
            <xs:enumeration value="16"/>
            <xs:enumeration value="17"/>
            <xs:enumeration value="18"/>
            <xs:enumeration value="19"/>
            <xs:enumeration value="20"/>
            <xs:enumeration value="21"/>
            <xs:enumeration value="22"/>
            <xs:enumeration value="23"/>
            <xs:enumeration value="24"/>
            <xs:enumeration value="25"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="JobOutputOptions" type="JobOutputOptionsType" minOccurs="0"/>
      <xs:element name="ModelOffset" type="ModelOffsetType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
element Root/Settings/InformFileFormats

- **InformFileFormatsType**
  - **attributes**
    - **MasterandSubs**
      - This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:
      - MasterName -- obsolete -- left in backwards compatibility - do not modify
      - MaxLines
        - Set from 100 to 950.
        - Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.
      - Example use:
        - InformFileFormats
        - MasterandSubs="False"
        - MasterName="gmaster"
        - MaxLines="900"
        - CallSub="M10"/
      - CallSub
        - Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.
        - Example use:
          - InformFileFormats
          - MasterandSubs="False"
          - MasterName="gmaster"
          - MaxLines="750"
          - CallSub="M10"/
      - ReturnSub
        - Creates a RET statement in the current Robot Job when a specific M or G code is
        - Example use:
          - InformFileFormats
          - MasterandSubs="False"
          - MasterName="gmaster"
          - MaxLines="750"
          - CallSub="M11"/
<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>annotation documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MasterandSubs</td>
<td>xs:string</td>
<td>required</td>
<td></td>
<td></td>
<td></td>
<td>This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:</td>
</tr>
<tr>
<td></td>
<td>MasterName</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MasterName -- obsolete – left in backwards compatibility - do not modify</td>
</tr>
<tr>
<td></td>
<td>MaxLines</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MaxLines Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.</td>
</tr>
</tbody>
</table>
|             | CallSub      |           |        |         |       | Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="900" CallSub="M10"/ CallSub Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This
function can be used for a number of functions.

Example use:
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="750"
CallSub="M10"

ReturnSub
Creates a RET statement in the current Robot Job when a specific M or G code is

Example use:
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="750"
CallSub="M11"

<table>
<thead>
<tr>
<th>MasterName</th>
<th>xs:string</th>
<th>optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxLines</td>
<td>xs:short</td>
<td>required</td>
</tr>
<tr>
<td>CallSub</td>
<td>xs:string</td>
<td>optional</td>
</tr>
<tr>
<td>ReturnSub</td>
<td>xs:string</td>
<td>optional</td>
</tr>
</tbody>
</table>

source <xs:element name="InformFileFormats" type="InformFileFormatsType"/>

element Root/Settings/ToolNumber

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="ToolNumber" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:int</td>
</tr>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>content simple</td>
</tr>
<tr>
<td></td>
<td>default 0</td>
</tr>
</tbody>
</table>
<xs:element name="ToolNumber" default="0">
  <xs:simpleType>
    <xs:restriction base="xs:int">
      <xs:enumeration value="0"/>
      <xs:enumeration value="1"/>
      <xs:enumeration value="2"/>
      <xs:enumeration value="3"/>
      <xs:enumeration value="4"/>
      <xs:enumeration value="5"/>
      <xs:enumeration value="6"/>
      <xs:enumeration value="7"/>
      <xs:enumeration value="8"/>
      <xs:enumeration value="9"/>
      <xs:enumeration value="10"/>
      <xs:enumeration value="11"/>
      <xs:enumeration value="12"/>
      <xs:enumeration value="13"/>
      <xs:enumeration value="14"/>
      <xs:enumeration value="15"/>
      <xs:enumeration value="16"/>
      <xs:enumeration value="17"/>
      <xs:enumeration value="18"/>
      <xs:enumeration value="19"/>
      <xs:enumeration value="20"/>
      <xs:enumeration value="21"/>
      <xs:enumeration value="22"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
### element Root/Settings/JobOutputOptions

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram of JobOutputOptions" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>JobOutputOptionsType</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>isRef</td>
</tr>
<tr>
<td>minOccurs</td>
</tr>
<tr>
<td>maxOcc</td>
</tr>
<tr>
<td>content</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>children</th>
</tr>
</thead>
<tbody>
<tr>
<td>JobType</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;xs:element name=&quot;JobOutputOptions&quot; type=&quot;JobOutputOptionsType&quot; minOccurs=&quot;0&quot;/&gt;</td>
</tr>
</tbody>
</table>

### element Root/Settings/ModelOffset

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram of ModelOffset" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>ModelOffsetType</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>properties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;xs:element name=&quot;ModelOffset&quot; type=&quot;ModelOffsetType&quot; minOccurs=&quot;0&quot;/&gt;</td>
</tr>
</tbody>
</table>
properties isRef 0
minOcc 0
maxOcc 1
content complex

children ModelOffsetX ModelOffsetY ModelOffsetZ ModelAngleRx ModelAngleRy ModelAngleRz

source
<xs:element name="ModelOffset" type="ModelOffsetType" minOccurs="0"/>

element Rx

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>xs:double</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>element ThreeDOF</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;Rx&quot; type=&quot;xs:double&quot;/&gt;</td>
</tr>
</tbody>
</table>

element Ry

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>xs:double</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>element ThreeDOF</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;Ry&quot; type=&quot;xs:double&quot;/&gt;</td>
</tr>
</tbody>
</table>

element Rz

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>xs:double</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>element ThreeDOF</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;Rz&quot; type=&quot;xs:double&quot;/&gt;</td>
</tr>
</tbody>
</table>

element ThreeDOF
### ThreeDOF

Diagram:
![Diagram of ThreeDOF](image)

**Properties**
- Content: Complex

**Children**
- Rx
- Ry
- Rz

**Used by**
- Element: **Root/Settings**

**Annotation**
Use this values to set the last length of the tool to be perpendicular to the part coordinate system. Generally this will be 180, 0, X Where X will be -90, 0, 90, 180, 270 degrees.

**Source**
```xml
<xs:element name="ThreeDOF">
  <xs:annotation>
    <xs:documentation>Use this values to set the last length of the tool to be perpendicular to the part coordinate system. Generally this will be 180, 0, X Where X will be -90, 0, 90, 180, 270 degrees.  </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Rx"/>
      <xs:element ref="Ry"/>
      <xs:element ref="Rz"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

### TimeUnits

Diagram:
![Diagram of TimeUnits](image)

**Type**
- Restriction of xs:string

**Properties**
- Content: Simple

**Used by**
- Element: **Units**

**Facets**
- Enumeration: Minutes
- Enumeration: Seconds
This element, and its sub elements specify the Length, Angle, and Time Units used in this XML File.
diagram

**InformFileFormatsType**

This element controls the form of the robot jobs that are created. The following Attributes are used:

- **MasterName** – obsolete – left in backwards compatibility - do not modify
- **MaxLines**
  - Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

Example use:

```
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="900"
CallSub="M10"
```

**CallSub**

Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

Example use:

```
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="750"
CallSub="M10"
```

**ReturnSub**

Creates a RET statement in the current Robot Job when a specific M or G code is found. This function can be used for a number of functions.

Example use:

```
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="750"
CallSub="M11"
```

**attributes**

**MasterandSubs**

This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:

- **MasterName** – obsolete – left in backwards compatibility - do not modify
- **MaxLines**
  - Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

Example use:

```
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="900"
CallSub="M10"
```

CallSub

Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

Example use:

```
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="900"
CallSub="M10"
```

ReturnSub

Creates a RET statement in the current Robot Job when a specific M or G code is found. This function can be used for a number of functions.

Example use:

```
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="750"
CallSub="M11"
```

**MasterName**
This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MasterandSubs</td>
<td>xs:string</td>
<td>required</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MaxLines</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CallSub</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ReturnSub</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**MasterName** -- obsolete – left in backwards compatibility - do not modify

**MaxLines**
Set from 100 to 950.
Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

Example use:
InformFileFormats
MasterandSubs="False"
MasterName="gmaster"
MaxLines="900" CallSub="M10"/

**CallSub**
Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used...
This element controls the form of the robot jobs that are created. The following Attributes are used:

MasterName  -- obsolete – left in backwards compatibility - do not modify

MaxLines  
Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="900" CallSub="M10"/

CallSub  
Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M11"/

ReturnSub  
Creates a RET statement in the current Robot Job when a specific M or G code is

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M11"/
<xs:complexType name="InformFileFormatsType">
  <xs:annotation>
    <xs:documentation>This element controls the form of the robot jobs that are created. The following Attributes are used:

    MasterName   -- obsolete -- left in backwards compatibility - do not modify

    MaxLines
    Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

    Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="900" CallSub="M10"/

    CallSub
    Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

    Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M10"/

    ReturnSub
    Creates a RET statement in the current Robot Job when a specific M or G code is

    Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M11"/
  </xs:documentation>
</xs:annotation>
  <xs:attribute name="MasterandSubs" type="xs:string" use="required">
    <xs:annotation>
      <xs:documentation>This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:

      MasterName   -- obsolete -- left in backwards compatibility - do not modify

      MaxLines
      Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

      Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="900" CallSub="M10"/

      CallSub
      Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

      Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M11"/
    </xs:documentation>
  </xs:attribute>
</xs:complexType>
MaxLines="750" CallSub="M10"/

ReturnSub
Creates a RET statement in the current Robot Job when a specific M or G code is

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M11"/

attribute InformFileFormatsType/@MasterandSubs

<table>
<thead>
<tr>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>use required</td>
</tr>
</tbody>
</table>

annotation
documentation
This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:

MasterName -- obsolete – left in backwards compatibility - do not modify

MaxLines
Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="900" CallSub="M10"/

CallSub
Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M10"/

ReturnSub
Creates a RET statement in the current Robot Job when a specific M or G code is

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M11"/
This element is used to control the form of the Robot Inform jobs that are created. This is done through the set of Element Attributes that are defined below:

MasterName -- obsolete – left in backwards compatibility - do not modify

MaxLines
Set from 100 to 950. Indicates the maximum number of positions functions that will go into a robot job or subroutine before the next subroutine is created.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="900" CallSub="M10"/

CallSub
Creates a Call Job instruction in the current robot job and when a specific M or G Code is found. This function can be used for a number of functions.

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M10"/

ReturnSub
Creates a RET statement in the current Robot Job when a specific M or G code is

Example use: InformFileFormats MasterandSubs="False" MasterName="gmaster" MaxLines="750" CallSub="M11"/

attribute InformFileFormatsType/@MasterName

<table>
<thead>
<tr>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>use optional</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:attribute name=&quot;MasterName&quot; type=&quot;xs:string&quot; use=&quot;optional&quot;/&gt;</td>
</tr>
</tbody>
</table>

attribute InformFileFormatsType/@MaxLines

<table>
<thead>
<tr>
<th>type</th>
<th>xs:short</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>use required</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:attribute name=&quot;MaxLines&quot; type=&quot;xs:short&quot; use=&quot;required&quot;/&gt;</td>
</tr>
</tbody>
</table>
attribute InformFileFormatsType/@CallSub

<table>
<thead>
<tr>
<th>property</th>
<th>type</th>
<th>properties</th>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>xs:string</td>
<td>isRef 0</td>
<td></td>
</tr>
<tr>
<td>properties</td>
<td>use optional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>source</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

attribute InformFileFormatsType/@ReturnSub

<table>
<thead>
<tr>
<th>property</th>
<th>type</th>
<th>properties</th>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>xs:string</td>
<td>isRef 0</td>
<td></td>
</tr>
<tr>
<td>properties</td>
<td>use optional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>source</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

complexType JobOutputOptionsType

<table>
<thead>
<tr>
<th>diagram</th>
<th>children</th>
<th>used by</th>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JobType</td>
<td>element Root/Settings/JobOutputOptions</td>
<td></td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:complexType name=&quot;JobOutputOptionsType&quot;&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="">xs:all</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element name=&quot;JobType&quot;&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="">xs:complexType</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="">xs:choice</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element name=&quot;PulseJobType&quot; default=&quot;True&quot; minOccurs=&quot;0&quot;&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="">xs:simpleType</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:restriction base=&quot;xs:string&quot;&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;True&quot;/&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;False&quot;/&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="">xs:restriction</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="">xs:simpleType</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="">xs:element</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element name=&quot;RectanJobType&quot; minOccurs=&quot;0&quot;&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="">xs:complexType</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="">xs:sequence</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element name=&quot;PositionConfiguration&quot; type=&quot;PositionConfigurationType&quot;/&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element name=&quot;RobotPosture&quot;/&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="">xs:complexType</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="">xs:all</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element name=&quot;FlipPosture&quot; default=&quot;0&quot;&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="">xs:simpleType</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:restriction base=&quot;xs:integer&quot;&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;0&quot;/&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
element JobOutputOptionsType/JobType

<table>
<thead>
<tr>
<th>diagram</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>properties</th>
<th>isRef 0</th>
<th>content complex</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>children</th>
<th>PulseJobType RectanJobType</th>
</tr>
</thead>
</table>

source

```xml
<xs:element name="JobType">
  <xs:complexType>
    <xs:choice>
      <xs:element name="PulseJobType" default="True" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="True"/>
            <xs:enumeration value="False"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="RectanJobType" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="PositionConfiguration" type="PositionConfigurationType"/>
            <xs:element name="RobotPosture">
              <xs:complexType>
                <xs:all>
                  <xs:element name="FlipPosture" default="0">
                    <xs:simpleType>
                      <xs:restriction base="xs:integer">
                        <xs:enumeration value="0"/>
                        <xs:enumeration value="1"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                  <xs:element name="UpperPosture" default="0">
                    <xs:simpleType>
                      <xs:restriction base="xs:integer">
                        <xs:enumeration value="0"/>
                        <xs:enumeration value="1"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                  <xs:element name="FrontPosture" default="0">
                    <xs:simpleType>
                      <xs:restriction base="xs:integer"/>
                    </xs:simpleType>
                  </xs:element>
                </xs:all>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>
</xs:element>
```
element JobOutputOptionsType/JobType/PulseJobType

diagram

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="PulseJobType" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>restriction of <code>xs:string</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>isRef 0</td>
</tr>
<tr>
<td>minOcc 0</td>
</tr>
<tr>
<td>maxOcc 1</td>
</tr>
<tr>
<td>content <code>simple</code></td>
</tr>
<tr>
<td>default <code>True</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>facets</th>
</tr>
</thead>
<tbody>
<tr>
<td>enumeration <code>True</code></td>
</tr>
<tr>
<td>enumeration <code>False</code></td>
</tr>
</tbody>
</table>
### PulseJobType

<table>
<thead>
<tr>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;xs:element name=&quot;PulseJobType&quot; default=&quot;True&quot; minOccurs=&quot;0&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:simpleType&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:restriction base=&quot;xs:string&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:enumeration value=&quot;True&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:enumeration value=&quot;False&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/xs:restriction&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/xs:simpleType&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/xs:element&gt;</code></td>
</tr>
</tbody>
</table>

### RectanJobType

**Diagram**

![Diagram of RectanJobType](image)

**Properties**
- `isRef`: 0
- `minOcc`: 0
- `maxOcc`: 1
- `content`: complex

**Children**
- `PositionConfiguration`
- `RobotPosture`

**Source**

<table>
<thead>
<tr>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;xs:element name=&quot;RectanJobType&quot; minOccurs=&quot;0&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:complexType&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:sequence&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element name=&quot;PositionConfiguration&quot; type=&quot;PositionConfigurationType&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element name=&quot;RobotPosture&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:complexType&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:all&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element name=&quot;FlipPosture&quot; default=&quot;0&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:simpleType&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:restriction base=&quot;xs:integer&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:enumeration value=&quot;0&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:enumeration value=&quot;1&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/xs:restriction&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/xs:simpleType&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/xs:element&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element name=&quot;UpperPosture&quot; default=&quot;0&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:simpleType&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:restriction base=&quot;xs:integer&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:enumeration value=&quot;0&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:enumeration value=&quot;1&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/xs:restriction&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/xs:simpleType&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/xs:element&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:element name=&quot;FrontPosture&quot; default=&quot;0&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:simpleType&gt;</code></td>
</tr>
<tr>
<td><code>&lt;xs:restriction base=&quot;xs:integer&quot;&gt;</code></td>
</tr>
</tbody>
</table>
element JobOutputOptionsType/JobType/RectanJobType/PositionConfiguration

diagram

```
<xs:element name="PositionConfiguration" type="PositionConfigurationType"/>
```

type PositionConfigurationType

properties
- isRef 0
- content complex

children RobotFrame BaseFrame UserFrame

source
```
<xs:element name="PositionConfiguration" type="PositionConfigurationType"/>
```
element `JobOutputOptionsType/JobType/RectanJobType/RobotPosture`

```
<xs:element name="RobotPosture">
  <xs:complexType>
    <xs:all>
      <xs:element name="FlipPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="UpperPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="FrontPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="RPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
        ```
element `JobOutputOptionsType/JobType/RectanJobType/RobotPosture/FlipPosture`

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="FlipPosture" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of <code>xs:integer</code></td>
</tr>
</tbody>
</table>
| properties | isRef 0  
|          | content simple  
|          | default 0  |
| facets  | enumeration 0  
|          | enumeration 1  |
| source  | <xs:element name="FlipPosture" default="0">  
|          | <xs:simpleType>  
|          | <xs:restriction base="xs:integer">  
|          | <xs:enumeration value="0"/>  
|          | <xs:enumeration value="1"/>  
|          | </xs:restriction>  
|          | </xs:simpleType>  
|          | </xs:element>  |

element `JobOutputOptionsType/JobType/RectanJobType/RobotPosture/UpperPosture`

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="UpperPosture" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of <code>xs:integer</code></td>
</tr>
</tbody>
</table>
| properties | isRef 0  
|          | content simple  
|          | default 0  |
| facets | enumeration 0  
|        | enumeration 1  |

source

```
<xs:element name="UpperPosture" default="0">
    <xs:simpleType>
        <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

---

**element** JobOutputOptionsType/JobType/RectanJobType/RobotPosture/FrontPosture

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:integer</th>
</tr>
</thead>
</table>
| properties | isRef 0  
|           | content simple 
|           | default 0                 |

| facets | enumeration 0  
|        | enumeration 1  |

source

```
<xs:element name="FrontPosture" default="0">
    <xs:simpleType>
        <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

---

**element** JobOutputOptionsType/JobType/RectanJobType/RobotPosture/RPosture

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:integer</th>
</tr>
</thead>
</table>
| properties | isRef 0  
|           | content simple 
|           | default 0                 |

| facets | enumeration 0  
|        | enumeration 1  |
### RPosture

**Source**

```xml
<xs:element name="RPosture" default="0">
    <xs:simpleType>
        <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

**Element** `JobOutputOptionsType/JobType/RectanJobType/RobotPosture/TPosture`

**Diagram**

![Diagram of TPosture]

**Type**

Restriction of `xs:integer`

**Properties**

- `isRef`: 0
- `content`: simple
- `default`: 0

**Facets**

- Enumeration 0
- Enumeration 1

**Source**

```xml
<xs:element name="TPosture" default="0">
    <xs:simpleType>
        <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

### ModelOffsetType

**complexType**

**Diagram**

![Diagram of ModelOffsetType]

**Children**

- `ModelOffsetX`
- `ModelOffsetY`
- `ModelOffsetZ`
- `ModelAngleRx`
- `ModelAngleRy`
- `ModelAngleRz`
used by  element Root/Settings/ModelOffset

source

```xml
<xs:complexType name="ModelOffsetType">
  <xs:sequence>
    <xs:element name="ModelOffsetX" type="xs:double" default="0.0" minOccurs="0"/>
    <xs:element name="ModelOffsetY" type="xs:double" default="0.0" minOccurs="0"/>
    <xs:element name="ModelOffsetZ" type="xs:double" default="0.0" minOccurs="0"/>
    <xs:element name="ModelAngleRx" type="xs:double" default="0.0" minOccurs="0"/>
    <xs:element name="ModelAngleRy" type="xs:double" default="0.0" minOccurs="0"/>
    <xs:element name="ModelAngleRz" type="xs:double" default="0.0" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

---

**element ModelOffsetType/ModelOffsetX**

diagram

```
+-------------------+
| ModelOffsetX      |
+-------------------+
```

type xs:double

properties

- isRef 0
- minOccurs 0
- maxOccurs 1
- content simple
- default 0.0

source

```xml
<xs:element name="ModelOffsetX" type="xs:double" default="0.0" minOccurs="0"/>
```

---

**element ModelOffsetType/ModelOffsetY**

diagram

```
+-------------------+
| ModelOffsetY      |
+-------------------+
```

type xs:double

properties

- isRef 0
- minOccurs 0
- maxOccurs 1
- content simple
- default 0.0

source

```xml
<xs:element name="ModelOffsetY" type="xs:double" default="0.0" minOccurs="0"/>
```

---

**element ModelOffsetType/ModelOffsetZ**

diagram

```
+-------------------+
| ModelOffsetZ      |
+-------------------+
```

type xs:double
| properties | isRef 0  
|            | minOcc 0  
|            | maxOcc 1  
|            | content simple  
|            | default 0.0  
| source     | `<xs:element name="ModelOffsetZ" type="xs:double" default="0.0" minOccurs="0"/>`  

**element ModelOffsetType/ModelAngleRx**

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
</table>
| type    | `xs:double`  
| properties | isRef 0  
|            | minOcc 0  
|            | maxOcc 1  
|            | content simple  
|            | default 0.0  
| source     | `<xs:element name="ModelAngleRx" type="xs:double" default="0.0" minOccurs="0"/>`  

**element ModelOffsetType/ModelAngleRy**

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
</table>
| type    | `xs:double`  
| properties | isRef 0  
|            | minOcc 0  
|            | maxOcc 1  
|            | content simple  
|            | default 0.0  
| source     | `<xs:element name="ModelAngleRy" type="xs:double" default="0.0" minOccurs="0"/>`  

**element ModelOffsetType/ModelAngleRz**

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
</table>
| type    | `xs:double`  
| properties | isRef 0  
|            | minOcc 0  
|            | maxOcc 1  
|            | content simple  
|            | default 0.0  
| source     | `<xs:element name="ModelAngleRz" type="xs:double" default="0.0" minOccurs="0"/>`
**complexType PositionConfigurationType**

| diagram |  
| --- | --- |
|  
|  
|  
|  
|  
|  
| PositionConfigurationType | RobotFrame | BaseFrame | UserFrame |

**children**

- RobotFrame
- BaseFrame
- UserFrame

**used by**
element `JobOutputOptionsType/JobType/RectanJobType/PositionConfiguration`

**source**

```xml
<xs:complexType name="PositionConfigurationType">
  <xs:choice>
    <xs:element name="RobotFrame" type="xs:string" default="True" minOccurs="0"/>
    <xs:element name="BaseFrame" default="False" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
          <xs:enumeration value="False"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="UserFrame" default="1" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:integer">
          <xs:totalDigits value="2"/>
          <xs:fractionDigits value="0"/>
          <xs:minInclusive value="0"/>
          <xs:maxInclusive value="25"/>
          <xs:pattern value="\d{1,2}"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:choice>
</xs:complexType>
```

**element PositionConfigurationType/RobotFrame**

| diagram |  
| --- | --- |
|  
|  
|  
|  
|  
|  
| RobotFrame |

**type**

`xs:string`
element PositionConfigurationType/BaseFrame

```xml
<xs:element name="BaseFrame" default="False" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="True"/>
      <xs:enumeration value="False"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

element PositionConfigurationType/UserFrame

```xml
<xs:element name="UserFrame" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:totalDigits value="2"/>
      <xs:fractionDigits value="0"/>
      <xs:pattern value="\d{1,2}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```
<xs:element name="UserFrame" default="1" minOccurs="0">
    <xs:simpleType>
        <xs:restriction base="xs:integer">
            <xs:totalDigits value="2"/>
            <xs:fractionDigits value="0"/>
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="25"/>
            <xs:pattern value="\d{1,2}"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
## Schema XMLtoJBISchemaV12.xsd

### Elements

- Add
- AngleSystem
- AngleUnits
- ArmConfig
- AxesPulsePosition
- AxesType
- B
- CallJob
- Comment
- Const
- DefaultMotionType
- DefaultVelocity
- DestIndex
- DestNum
- DestType
- DistVar
- Divide
- ExternalAxes
- InputPositionData
- JobData
- JobOutputOptions
- Joint
- L
- LengthUnits
- Linear
- MotionType
- MoveTag
- OutputOff
- OutputOn
- Position
- PosLevel
- R
- ReferenceModel
- RefFrame
- Root
- Rx
- Ry
- Rz
- S
- SetPulseConfig
- Settings
- SetVar
- SourceNum
- SourceType
- Sub
- T
- ThreeDOF
- Timer
- TimeUnits
- ToolNumber
- U

### Complex types

- BaseStationType
- ConvertType
- GetCurPos
- InputPositionDataType
- JobOutputOptionsType
- ModelOffsetType
- PositionConfigurationType
- PositionConfigurationPulseType
- RotatingStationJobDataPositionType
- RotatingStationPositionType
- RotatingStationSetupType
element **Add**

- **type**: `SetVar`
- **properties**: content `complex`
- **children**: `Const DestType DestIndex SourceNumSourceType DestNum DestType`
- **used by**: element **JobData**
- **source**: `<xs:element name="Add" type="SetVar"/>

**Diagram**

```
  Add
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
  |   |
This element specifies the reference system for position orientation angles. Values are YPR and Euler. YPR is the standard MotoSimEG angle system. Euler means that the angle system is specified in terms of the Euler Angle system. This is a system that is used in the aircraft industry.

```xml
<xs:element name="AngleSystem">
  <xs:annotation>
    <xs:documentation>This element specifies the reference system for position orientation angles. Values are YPR and Euler. YPR is the standard MotoSimEG angle system. Euler means that the angle system is specified in terms of the Euler Angle system. This is a system that is used in the aircraft industry.</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="YPR"/>
      <xs:enumeration value="Euler"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

element **AngleUnits**

diagram

<table>
<thead>
<tr>
<th>facet</th>
<th>enumeration Degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>enumeration Radians</td>
</tr>
</tbody>
</table>

type   restriction of xs:string

properties  content simple

used by  element **Units**

source  

```xml
<xs:element name="AngleUnits">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Degrees"/>
      <xs:enumeration value="Radians"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

element **ArmConfig**
Specifies the robot initial starting position and form. This position is specified in joint coordinates and will set the initial pose (elbow up/down, etc.) of the robot.

This position can be set by moving the simulated robot to the part coordinate system 0,0,0 location and setting the individual axes to archive the desired pose. The values can be copied from MotoSimEG and pasted in the appropriate joint location.

Note: You can specify up to 21 unique robot configurations.
Note: The first set of Arm Configuration values are automatically selected.
Note: Other Configurations can be selected by the SetPulseConfig commands.
Specifies the robot initial starting position and form. This position is specified in joint coordinates and will set the initial pose (elbow up/down, etc.) of the robot.

This position can be set by moving the simulated robot to the part coordinate system 0,0,0 location and setting the individual axes to achieve the desired pose. The values can be copied from MotoSimEG and pasted in the appropriate joint location.

Note: You can specify up to 21 unique robot configurations.

Note: The first set of Arm Configuration values are automatically selected.

Note: Other Configurations can be selected by the SetPulseConfig commands.

attribute ArmConfig/@JobName

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>use optional</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration default</td>
</tr>
</tbody>
</table>

source <xs:attribute name="JobName" use="optional">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="default"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>

element AxesPulsePosition
**element AxesPulsePosition**

<table>
<thead>
<tr>
<th>type</th>
<th>xs:integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>element Settings</td>
</tr>
<tr>
<td>annotation</td>
<td>documentation Used with the ExternalAxes is set to True. When ExternalAxes is True, this element stores the static external Axis pulse count.</td>
</tr>
</tbody>
</table>
| source     | <xs:element name="AxesPulsePosition" type="xs:integer">
  <xs:annotation>
    <xs:documentation>Used with the ExternalAxes is set to True. When ExternalAxes is True, this element stores the static external Axis pulse count.</xs:documentation>
  </xs:annotation>
</xs:element> |

**element AxesType**

<table>
<thead>
<tr>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>element Settings</td>
</tr>
<tr>
<td>annotation</td>
<td>documentation This element is used with ExternalAxes Element. When the External Axes is True, this element tells the robot the type of External Axes. Currently only &quot;Robot&quot; AxesType is supported.</td>
</tr>
</tbody>
</table>
| source     | <xs:element name="AxesType" type="xs:string">
  <xs:annotation>
    <xs:documentation>This element is used with ExternalAxes Element. When the External Axes is True, this element tells the robot the type of External Axes. Currently only "Robot" AxesType is supported.</xs:documentation>
  </xs:annotation>
</xs:element> |

**element B**

<p>| diagram | |
|---------| |</p>
<table>
<thead>
<tr>
<th>type</th>
<th>xs:long</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>element ArmConfig</td>
</tr>
<tr>
<td>source</td>
<td><code>&lt;xs:element name=&quot;B&quot; type=&quot;xs:long&quot;/&gt;</code></td>
</tr>
</tbody>
</table>

**element CallJob**

- **diagram**
  
- **type** restriction of xs:string
- **properties** content simple
- **used by** element JobData
- **facets** minLength 1, maxLength 8
- **annotation documentation**
  XML to JBI Command. Creates an inform command with the form: CALL JOB:xxxxxxxxxx where xxxxxxxxx is the job name.
- **source**
  ```xml
  <xs:element name="CallJob">
    <xs:annotation>
      <xs:documentation>XML to JBI Command. Creates an inform command with the form:
      CALL JOB:xxxxxxxxxx where xxxxxxxxx is the job name.</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="8"/>
        <xs:minLength value="1"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  ```

**element Comment**

- **diagram**
  
- **type** xs:string
- **properties** content simple
- **used by** element JobData
- **annotation documentation**
  Add a ' mark before the message. Controller will ignore these instructions.
element `Const`

```
<xs:element name="Const"/>
```

```
<xs:annotation>
  <xs:documentation>
    Const
  </xs:documentation>
</xs:annotation>
```

used by: complexType `SetVar`

```
source <xs:element name="Const"/>
```

---

element `DefaultMotionType`

```
<xs:element name="DefaultMotionType">
  <xs:annotation>
    <xs:documentation>
      When a position command does not include a MotionType command, the DefaultMotionType is used.
    </xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Linear"/>
      <xs:enumeration value="Circular"/>
      <xs:enumeration value="ExternalRef"/>
      <xs:enumeration value="Spline"/>
      <xs:enumeration value="Inc"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

```
source <xs:element name="DefaultMotionType">
  <xs:annotation>
    <xs:documentation>
      When a position command does not include a MotionType command, the DefaultMotionType is used.
    </xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Linear"/>
      <xs:enumeration value="Circular"/>
      <xs:enumeration value="ExternalRef"/>
      <xs:enumeration value="Spline"/>
      <xs:enumeration value="Inc"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

---

```
source <xs:element name="Comment" type="xs:string">
  <xs:annotation>
    <xs:documentation>Add a ' mark before the message. Controller will ignore these instructions.</xs:documentation>
  </xs:annotation>
</xs:element>
```

---

element `DefaultVelocity`
Default Velocity is the velocity that is used when a position command does not include a velocity command.

```xml
<xs:element name="DefaultVelocity">
  <xs:annotation>
    <xs:documentation>Default Velocity is the velocity that is used when a position command does not include a velocity command.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Linear" minOccurs="0"/>
      <xs:element ref="Joint" minOccurs="0"/>
      <xs:element name="ExternalRef" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:decimal">
            <xs:fractionDigits value="10"/>
            <xs:totalDigits value="16"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Circular" type="xs:anySimpleType" minOccurs="0"/>
      <xs:element name="Spline" type="xs:anySimpleType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**Element DefaultVelocity/ExternalRef**

**Diagram**

**Type**

restriction of `xs:decimal`

**Properties**

- isRef: 0
- minOccurs: 0
- maxOccurs: 1
- content: simple
element DefaultVelocity/Circular

diagram

```
Circular
```

type xs:anySimpleType

properties
- isRef: 0
- minOccurs: 0
- maxOccurs: 1
- content: simple

source
```
<xs:element name="Circular" type="xs:anySimpleType" minOccurs="0"/>
```

element DefaultVelocity/Spline

diagram

```
Spline
```

type xs:anySimpleType

properties
- isRef: 0
- minOccurs: 0
- maxOccurs: 1
- content: simple

source
```
<xs:element name="Spline" type="xs:anySimpleType" minOccurs="0"/>
```

element DestIndex

diagram

```
DestIndex
```

type xs:integer

properties
- content: simple

used by complexTypes SetPosElmnt SetVar

source
```
<xs:element name="DestIndex" type="xs:integer"/>
```

element DestNum

diagram

```
DestNum
```

source
```
<xs:element name="DestIndex" type="xs:integer"/>
```
### element DestType

**Diagram:**
![DestType diagram](image)

<table>
<thead>
<tr>
<th>type</th>
<th><code>xs:anyType</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content: complex, mixed: true</td>
</tr>
<tr>
<td>used by</td>
<td>complexType <code>SetVar</code></td>
</tr>
<tr>
<td>attributes</td>
<td>Name</td>
</tr>
<tr>
<td>source</td>
<td><code>&lt;xs:element name=&quot;DestType&quot; type=&quot;xs:anyType&quot;/&gt;</code></td>
</tr>
</tbody>
</table>

### element DistVar

**Diagram:**
![DistVar diagram](image)

<table>
<thead>
<tr>
<th>type</th>
<th><code>xs:anySimpleType</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content: simple, default: 1</td>
</tr>
<tr>
<td>used by</td>
<td>complexType <code>Search</code></td>
</tr>
<tr>
<td>source</td>
<td><code>&lt;xs:element name=&quot;DistVar&quot; type=&quot;xs:anySimpleType&quot; default=&quot;1&quot;/&gt;</code></td>
</tr>
</tbody>
</table>

### element Divide
### element `SetVar`

<table>
<thead>
<tr>
<th>property</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td><code>SetVar</code></td>
</tr>
<tr>
<td>properties</td>
<td>content complex</td>
</tr>
<tr>
<td>children</td>
<td><code>Const DestType DestIndex SourceNum SourceType DestNum DestType</code></td>
</tr>
<tr>
<td>used by</td>
<td>element <code>JobData</code></td>
</tr>
<tr>
<td>source</td>
<td><code>&lt;xs:element name=&quot;Divide&quot; type=&quot;SetVar&quot;/&gt;</code></td>
</tr>
</tbody>
</table>

### element `ExternalAxes`

<table>
<thead>
<tr>
<th>property</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of <code>xs:string</code></td>
</tr>
<tr>
<td>properties</td>
<td>content <code>simple</code></td>
</tr>
<tr>
<td>used by</td>
<td>element <code>Settings</code></td>
</tr>
<tr>
<td>facets</td>
<td>enumeration <code>False</code></td>
</tr>
<tr>
<td></td>
<td>enumeration <code>True</code></td>
</tr>
<tr>
<td>annotation</td>
<td>documentation</td>
</tr>
<tr>
<td></td>
<td>This element is set to True when the Robot is equipped with an external Axis. Otherwise it is set to False. Current Converter only supports a static external axis.</td>
</tr>
</tbody>
</table>
source
<xs:element name="ExternalAxes">
  <xs:annotation>
    <xs:documentation>This element is set to True when the Robot is equipped with an external Axis. Otherwise it is set to False.
    Current Converter only supports a static external axis. </xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="False"/>
      <xs:enumeration value="True"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

element InputPositionData

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>InputPositionDataType</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content complex</td>
</tr>
<tr>
<td>children</td>
<td>InputPositionDataPulse InputPositionDataTypeRectan</td>
</tr>
<tr>
<td>used by</td>
<td>element Settings</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;InputPositionData&quot; type=&quot;InputPositionDataType&quot;/&gt;</td>
</tr>
</tbody>
</table>

element JobData

diagram

Add a ‘ mark before the message. Controller will ignore these instructions.

OutputOn

XML to JBI Instruction.
Creates a Inform Instruction:
DOUT OT(#X) ON

OutputOff

XML to JBI Instruction.
Creates a Inform Instruction:
DOUT OT(#X) OFF

Position

Use this to set the Position of the Robot.

Comment

Add a note or description.
**Position**

This is the element that specifies the move positions. Velocities, motion types and other ancillary position information.

**Timer**

XML to JBI Instruction. Creates an inform command with the following form: TIMER T=X where X is the number of seconds to delay.

**GenInform**

GenInform is an abbreviation for General Inform. This is a pass through command that is not syntax checked. This is the command allows users to use any advanced inform command.

**SetVar**

SetVar is an XML to JBI Instruction that allows the setting of Robot Variables. These variables may be B, I, D or R variables.

P variables are set with SetPosElmnt.

This element may have one of two forms. In form one, you specify the Variable Type and Index and provide the variable value with a "Const" element. In Form 2 you specify the source and destination variable types and indices.

**SetPosElmnt**

SetPosElmnt is the XML to JBI command that allows the user to set an element of a P Variable. There are two forms for this command. Either a constant can be

**GetCurPos**

Pleases the current robot position in the P variable whose index is given in the DestNum Tag. Data Type can be Pulse or
whose index is given in the
DestNum Tag.
DataType can be Pulse or
Cart.
When DataType is Cart the
return value is in Robot
Frame Cart. Coordinates.

NOTE: $PX000 always has
Pulse Values and $PX001
always returns Cart.
Coordinates.

Example 1:
XML Syntax of
DestNum = 5 and DataType
= Cart will create an Inform
Instruction of
GETS PX005 $PX001

Example 2:
XML Syntax of
DestNum = 3 and DataType
= Pulse will create an
Inform Instruction of
GETS PX003 $PX000

CallJob

XML to JBI Command.
Creates an inform command
with the form:
CALL JOB1xxxxxxxx where
xxxxxxxx is the job name.

SetPulseConfig

This command allows you to
select the desired robot form
or starting configuration.
This command causes a
PUTP(POS[100X]) command
in the intermediate
JobName.in file. Where X is
the Index Value.
The index is used to select
from the ArmConfig
commands in the Settings
Section of the XML file.

NOTE:
1. Index must be in the ran
This is the section of the file that contains the instructions that will be converted into Inform Instructions.

This is the section of the file that contains the instructions that will be converted into Inform Instructions. This is the command that allows users to use any advanced inform command. GenInform is an abbreviation for General Inform. This is a pass through command that is not syntax checked.

SetPosElmnt is the XML to JBI command that allows the user to set an element of a P Variable. There are two forms for this command. Either a constant can be

Pleases the current robot position in the P variable whose index is given in the DestNum Tag. DataType can be Pulse or Cart.
When DataTable is Cart the return value is in Robot Frame Cart. Coordinates.

NOTE: $PX000 always has Pulse Values and $PX001 always returns Cart. Coordinates.

Example 1:
XML Syntax of
DestNum = 5 and DataType = Cart will create an Inform Instruction of
GETS PX005 $PX001

Example 2:
XML Syntax of
DestNum = 3 and DataType = Pulse will create an Inform Instruction of
GETS PX003 $PX000

element **JobData/GenInform**

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td>annotation</td>
<td>GenInform is an abbreviation for General Inform. This is a pass through command that is not syntax checked. This is the command allows users to use any advanced inform command.</td>
</tr>
</tbody>
</table>
source
<source name="GenInform" type="xs:string" minOccurs="0" maxOccurs="unbounded">  
<annotation>
  <documentation>GenInform is an abbreviation for General Inform. This is a pass through command that is not syntax checked. This is the command allows users to use any advanced inform command. </documentation>
</annotation>
</source>

element JobData/SetPosElmnt

SetPosElmnt is the XML to JBI command that allow the user to set an element of a P Variable. There are two forms for this command. Either a constant can be set.

source
<source name="SetPosElmnt" type="SetPosElmnt" minOccurs="0" maxOccurs="unbounded">  
<annotation>
  <documentation>SetPosElmnt is the XML to JBI command that allow the user to set an element of a P Variable. There are two forms for this command. Either a constant can be set. </documentation>
</annotation>
</source>

element JobData/GetCurPos
diagram

GetCurPos

<table>
<thead>
<tr>
<th>type</th>
<th>GetCurPos</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td></td>
</tr>
<tr>
<td>isRef</td>
<td>0</td>
</tr>
<tr>
<td>minOcc</td>
<td>0</td>
</tr>
<tr>
<td>maxOcc</td>
<td>unbounded</td>
</tr>
<tr>
<td>content</td>
<td>complex</td>
</tr>
<tr>
<td>children</td>
<td>DestNum, DataType, RefFrame</td>
</tr>
<tr>
<td>annotation</td>
<td>documentation</td>
</tr>
</tbody>
</table>

Pleases the current robot position in the P variable whose index is given in the DestNum Tag. 
DataType can be Pulse or Cart. 
When DataType is Cart the return value is in Robot Frame Cart. Coordinates. 

NOTE: $PX000 always has Pulse Values and $PX001 allways returns Cart. Coordinates. 

Example 1:
XML Syntax of 
DestNum = 5 and DataType = Cart will create an Inform Instruction of 
GETS PX005 $PX001 

Example 2:
XML Syntax of 
DestNum = 3 and DataType = Pulse will create an Inform Instruction of 
GETS PX003 $PX000
Pleases the current robot position in the P variable whose index is given in the DestNum Tag.
DataType can be Pulse or Cart.
When DataType is Cart the return value is in Robot Frame Cart. Coordinates.

NOTE: $PX000 always has Pulse Values and $PX001 allways returns Cart. Coordinates.

Example 1:
XML Syntax of DestNum = 5 and DataType = Cart will create an Inform Instruction of GETS PX005 $PX001

Example 2:
XML Syntax of DestNum = 3 and DataType = Pulse will create an Inform Instruction of GETS PX003 $PX000

XXxFillIn
### element JobData/Convert

**Diagram**

![Diagram of Convert](image)

**Type**

`${ConvertType}`

**Properties**

- `isRef`: 0
- `minOcc`: 0
- `maxOcc`: unbounded
- `content`: complex

**Children**

- `SourceNum`
- `DestNum`
- `RefFrame`

**Source**

```xml
<xs:element name="Convert" type="ConvertType" minOccurs="0" maxOccurs="unbounded"/>
```

### element JobData/Gete

**Diagram**

![Diagram of Gete](image)

**Properties**

- `isRef`: 0
- `minOcc`: 0
- `maxOcc`: unbounded

**Source**

```xml
<xs:element name="Gete" minOccurs="0" maxOccurs="unbounded"/>
```

### element JobData/Sete

**Diagram**

![Diagram of Sete](image)

**Properties**

- `isRef`: 0
- `minOcc`: 0
- `maxOcc`: unbounded

**Source**

```xml
<xs:element name="Sete" minOccurs="0" maxOccurs="unbounded"/>
```

### element JobData/ShiftOn
**type** ShiftOnType

**properties**
- isRef: 0
- minOccurs: 0
- maxOccurs: unbounded
- content: complex

**children**
- SourceNum
- RefFrame

**source**
```
<xs:element name="ShiftOn" type="ShiftOnType" minOccurs="0" maxOccurs="unbounded"/>
```

**element** JobOutputOptions

**type** JobOutputOptionsType

**properties**
- content: complex

**children**
- JobType

**used by**
- element Settings

**source**
```
<xs:element name="JobOutputOptions" type="JobOutputOptionsType"/>
```

**element** Joint

**type** restriction of xs:integer

**properties**
- content: simple
- default: 20

**used by**
- element DefaultVelocity
facets
  totalDigits 3
  fractionDigits 0
  pattern \d{1,3}

source
<xs:element name="Joint" default="20">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:totalDigits value="3"/>
      <xs:pattern value="\d{1,3}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

element L

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>xs:long</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>element ArmConfig</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;L&quot; type=&quot;xs:long&quot;/&gt;</td>
</tr>
</tbody>
</table>

element LengthUnits

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>element Units</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration Inches enumeration Millimeters</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;LengthUnits&quot;&gt;</td>
</tr>
</tbody>
</table>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Inches"/>
      <xs:enumeration value="Millimeters"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

element Linear

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:decimal</th>
</tr>
</thead>
</table>
| properties | content simple  
|            | default 100.0 |
| used by   | element DefaultVelocity |
| facets    | totalDigits 16  
|           | fractionDigits 10  
|           | pattern \( \d{1,6} \).*\( \d{0,10} \) |
| source    | `<xs:element name="Linear" default="100.0">  
|           | <xs:simpleType>  
|           | <xs:restriction base="xs:decimal">  
|           | <xs:fractionDigits value="10"/>  
|           | <xs:totalDigits value="16"/>  
|           | <xs:pattern value="(\d{1,6})*\d{0,10}"/>  
|           | </xs:restriction>  
|           | </xs:simpleType>  
|           | </xs:element>` |

**element MotionType**

| diagram | ![Diagram](image) |
| type    | restriction of xs:string |
| properties | content simple  
|           | default Joint |
| used by  | element Position |
| facets   | maxLength 31  
|          | enumeration Joint  
|          | enumeration Linear  
|          | enumeration Circular  
|          | enumeration Inc  
|          | enumeration Spline  
|          | enumeration SynchLinear  
|          | enumeration SynchCircular |
| annotation | documentation  
|           | Specifies the Motion type to use to move the robot to this location. |
| source   | `<xs:element name="MotionType" default="Joint">  
|          | <xs:annotation>  
|          | <xs:documentation>Specifies the Motion type to use to move the robot to this location. </xs:documentation>  
|          | </xs:annotation>  
|          | </xs:simpleType>  
|          | <xs:restriction base="xs:string">  
|          | <xs:maxLength value="31"/>  
|          | <xs:enumeration value="Joint"/>  
|          | <xs:enumeration value="Linear"/>  
|          | <xs:enumeration value="Circular"/>  
|          | <xs:enumeration value="Inc"/>  
|          | <xs:enumeration value="Spline"/>  
|          | </xs:restriction>` |
**element MoveTag**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="MoveTag" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of <code>xs:string</code></td>
</tr>
<tr>
<td>properties</td>
<td>content simple default NWAIT</td>
</tr>
<tr>
<td>used by</td>
<td>element <code>Position</code></td>
</tr>
<tr>
<td>facets</td>
<td>enumeration NWAIT</td>
</tr>
</tbody>
</table>
| source | `<xs:element name="MoveTag" default="NWAIT">  
  <xs:simpleType>  
  <xs:restriction base="xs:string">  
  <xs:enumeration value="NWAIT"/>  
  </xs:restriction>  
  </xs:simpleType>  
</xs:element>` |

**element OutputOff**

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="OutputOff" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td><code>xs:byte</code></td>
</tr>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>element <code>JobData</code></td>
</tr>
</tbody>
</table>
| annotation | documentation  
XML: to JBI Instruction.  
Creates a Inform Instruction:  
DOUT OT(#X) OFF |
| source | `<xs:element name="OutputOff" type="xs:byte">  
  <xs:annotation>  
  <xs:documentation>XML: to JBI Instruction.  
Creates a Inform Instruction:  
DOUT OT(#X) OFF</xs:documentation>  
  </xs:annotation>  
</xs:element>` |

**element OutputOn**
Type: xs:byte

Properties: Content simple

Used by: Element JobData

Annotation documentation:
XML: to JBI Instruction.
Creates a Inform Instruction:
DOUT OT(#X) ON

Source:
<xs:element name="OutputOn" type="xs:byte">
  <xs:annotation>
    <xs:documentation>XML: to JBI Instruction.
    Creates a Inform Instruction:
    DOUT OT(#X) ON
  </xs:documentation>
</xs:element>

Element Position:

Diagram:

MotionType

Specifies the Motion type to use to move the robot to this location.

Cartesian coordinates can include XYZ or XYZRxRyRz. If Rx, Ry, and Rz are not specified, the default ThreeDOF Data will be used.

Robot positions can be in Cartesian Coordinates, Joint Coordinates or in a P Variable. P variable maybe in either Joint or Cartesian.
Position

This is the element that specifies the move positions, Velocities, motion types and other ancillary position information.

Velocity

RefFrame

This command is used to allow the robot to move until a variable is On or Off. Maximum Search distance is specified in the Const Tag. The Actual Search Distance in placed in the D Variable whose index is specified in the DistVar tag.

NOTE: Only works with IMOV commands.

Inform Syntax is:
SRCH RIN#{(x)} = y DIS=Z
Dw where:
x = Input
y = State
z = Const
w = DistVar

MoveTag

RotatingStationPosition

0...4
<table>
<thead>
<tr>
<th>properties</th>
<th>content: complex</th>
</tr>
</thead>
<tbody>
<tr>
<td>children</td>
<td>MotionType X Y Z Ry Rx S L U B T PosVar Velocity PosLevel RefFrame Tool Form Search MoveTag RotatingStationPosition</td>
</tr>
<tr>
<td>used by</td>
<td>element JobData</td>
</tr>
<tr>
<td>annotation documentation</td>
<td>This is the element that specifies the move positions, Velocities, motion types and other ancillary position information.</td>
</tr>
</tbody>
</table>

```xml
<xs:element name="Position">
   <xs:annotation>
      <xs:documentation>This is the element that specifies the move positions, Velocities, motion types and other ancillary position information.</xs:documentation>
   </xs:annotation>
   <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
         <xs:element ref="MotionType" minOccurs="0"/>
         <xs:choice>
            <xs:annotation>
               <xs:documentation>Robot positions can be in Cartesian Coordinates, Joint Coordinates or in a P Variable. P variable maybe in either Joint or Cartesian. </xs:documentation>
            </xs:annotation>
            <xs:sequence minOccurs="0">
               <xs:element name="S" type="xs:short"/>
               <xs:element name="L" type="xs:short"/>
               <xs:element name="U" type="xs:short"/>
               <xs:element name="B" type="xs:short"/>
               <xs:element name="T" type="xs:short"/>
            </xs:sequence>
            <xs:sequence minOccurs="0">
               <xs:element name="PosVar" default="0">
                  <xs:simpleType>
                     <xs:restriction base="xs:integer">
                        <xs:fractionDigits value="0"/>
                        <xs:totalDigits value="3"/>
                     </xs:restriction>
                  </xs:simpleType>
               </xs:element>
            </xs:sequence>
         </xs:choice>
         <xs:choice>
            <xs:annotation>
               <xs:documentation>Cartesian coordinates can include XYZ or XYZRxRyRz. If Rx, Ry, and Rz are not specified, the default ThreeDOF Data will be used.</xs:documentation>
            </xs:annotation>
            <xs:sequence minOccurs="0">
               <xs:element ref="X" minOccurs="0"/>
               <xs:element ref="Y" minOccurs="0"/>
               <xs:element ref="Z" minOccurs="0"/>
               <xs:element ref="Ry" minOccurs="0"/>
               <xs:element ref="Rx" minOccurs="0"/>
               <xs:element ref="Rz" minOccurs="0"/>
            </xs:sequence>
         </xs:choice>
      </xs:choice>
      <xs:sequence minOccurs="0">
         <xs:element name="Velocity" minOccurs="0">
            <xs:simpleType>
               <xs:restriction base="xs:decimal"/>
            </xs:simpleType>
         </xs:element>
      </xs:sequence>
   </xs:choice>
</xs:element>
```
<xs:element name="Search" type="Search" minOccurs="0">
  <xs:annotation>
    <xs:documentation>This command is used to allow the robot to move until a variable is On or Off. Maximum Search distance is specified in the Const Tag. The Actual Search Distance in placed in the D Variable whose index is specified in the DistVar tag.

    NOTE: Only works with IMOV commands.
  </xs:annotation>
  <xs:element ref="MoveTag" minOccurs="0"/>
  <xs:element name="RotatingStationPosition" type="RotatingStationJobDataPositionType" minOccurs="0" maxOccurs="4"/>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>

<table>
<thead>
<tr>
<th>element</th>
<th>Position/S</th>
</tr>
</thead>
<tbody>
<tr>
<td>diagram</td>
<td><img src="image" alt="Diagram" /></td>
</tr>
<tr>
<td>type</td>
<td>xs:short</td>
</tr>
</tbody>
</table>
| properties | isRef 0  
| content  | simple  |
| used by | element ArmConfig |
| source  | <xs:element name="S" type="xs:short"/> |

<table>
<thead>
<tr>
<th>element</th>
<th>Position/L</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>diagram</th>
<th>type</th>
<th>xs:short</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef</td>
<td>0</td>
</tr>
<tr>
<td>content</td>
<td>simple</td>
<td></td>
</tr>
<tr>
<td>used by</td>
<td>element</td>
<td>ArmConfig</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;L&quot; type=&quot;xs:short&quot;/&gt;</td>
<td></td>
</tr>
</tbody>
</table>

**element Position/U**

<table>
<thead>
<tr>
<th>diagram</th>
<th>type</th>
<th>xs:short</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef</td>
<td>0</td>
</tr>
<tr>
<td>content</td>
<td>simple</td>
<td></td>
</tr>
<tr>
<td>used by</td>
<td>element</td>
<td>ArmConfig</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;U&quot; type=&quot;xs:short&quot;/&gt;</td>
<td></td>
</tr>
</tbody>
</table>

**element Position/B**

<table>
<thead>
<tr>
<th>diagram</th>
<th>type</th>
<th>xs:short</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef</td>
<td>0</td>
</tr>
<tr>
<td>content</td>
<td>simple</td>
<td></td>
</tr>
<tr>
<td>used by</td>
<td>element</td>
<td>ArmConfig</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;B&quot; type=&quot;xs:short&quot;/&gt;</td>
<td></td>
</tr>
</tbody>
</table>

**element Position/T**

<table>
<thead>
<tr>
<th>diagram</th>
<th>type</th>
<th>xs:short</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef</td>
<td>0</td>
</tr>
<tr>
<td>content</td>
<td>simple</td>
<td></td>
</tr>
<tr>
<td>used by</td>
<td>element</td>
<td>ArmConfig</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;T&quot; type=&quot;xs:short&quot;/&gt;</td>
<td></td>
</tr>
</tbody>
</table>

**element Position/PosVar**

<table>
<thead>
<tr>
<th>diagram</th>
<th>type</th>
<th>xs:short</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef</td>
<td>0</td>
</tr>
<tr>
<td>content</td>
<td>simple</td>
<td></td>
</tr>
</tbody>
</table>
### Position/Velocity

- **Type**: xs:decimal
- **Properties**:
  - isRef: 0
  - minOccurs: 0
  - maxOccurs: 1
  - content: simple
- **Facets**:
  - totalDigits: 16
  - fractionDigits: 6
- **Source**:
  ```xml
  <xs:element name="Velocity" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:decimal">
        <xs:totalDigits value="16"/>
        <xs:fractionDigits value="6"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  ```

### Position/Tool

- **Type**: xs:integer
- **Properties**:
  - isRef: 0
  - minOccurs: 0
  - maxOccurs: 1
  - content: simple
  - default: 0
- **Annotation**:
  Specifies the Tool Number to user for this move.

---

<table>
<thead>
<tr>
<th>Type</th>
<th>xs:integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>minOccurs 0</td>
</tr>
<tr>
<td></td>
<td>maxOccurs 1</td>
</tr>
<tr>
<td></td>
<td>content simple</td>
</tr>
<tr>
<td></td>
<td>default 0</td>
</tr>
<tr>
<td>Annotation</td>
<td>documentation</td>
</tr>
<tr>
<td></td>
<td>Specifies the Tool Number to user for this move.</td>
</tr>
</tbody>
</table>
### element `Tool`:

<table>
<thead>
<tr>
<th>source</th>
<th><code>&lt;xs:element name=&quot;Tool&quot; type=&quot;xs:integer&quot; default=&quot;0&quot; minOccurs=&quot;0&quot;&gt;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>&lt;xs:annotation&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:documentation&gt;Specifies the Tool Number to user for this move.&lt;/xs:documentation&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/xs:annotation&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/xs:element&gt;</code></td>
</tr>
</tbody>
</table>

### element `Position/Form`:

- **Diagram**: ![Diagram](image)
- **Type**: `xs:integer`
- **Properties**:
  - `isRef`: 0
  - `minOcc`: 0
  - `maxOcc`: 1
  - `content`: simple
- **Annotation**:
  - `documentation`: XXXFillIn
- **Source**:
  ```xml
  <xs:element name="Form" type="xs:integer" minOccurs="0">
  <xs:annotation>
    <xs:documentation>XXXFillIn</xs:documentation>
  </xs:annotation>
  </xs:element>
  ```

### element `Position/Search`:

- **Diagram**: ![Diagram](image)
- **Type**: `Search`
- **Text**:
  
  *This command is used to allow the robot to move until a variable is On or Off. Maximum Search distance is specified in the Const Tag. The Actual Search Distance in placed in the D Variable whose index is specified in the DistVar tag.*

  **NOTE:** Only works with IMOV commands.

  **Inform Syntax is:**
  
  `SRCH RIN#(x) = y DIS=Z`  
  `Dw where:`
  
  `x = Input`
  
  `y=State`
  
  `z=Const`
  
  `w=DistVar`
This command is used to allow the robot to move until a variable is On or Off. Maximum Search distance is specified in the Const Tag. The Actual Search Distance in placed in the D Variable whose index is specified in the DistVar tag.

NOTE: Only works with IMOV commands.

Inform Syntax is:
SRCH RIN#(x) = y DIS=Z Dw where:
x = Input
y=State
z=Const
w=DistVar

<sx:element name="Search" type="Search" minOccurs="0">
  <xs:annotation>
    <xs:documentation>This command is used to allow the robot to move until a variable is On or Off. Maximum Search distance is specified in the Const Tag. The Actual Search Distance in placed in the D Variable whose index is specified in the DistVar tag.
    
    NOTE: Only works with IMOV commands.
    
    Inform Syntax is:
    SRCH RIN#(x) = y DIS=Z Dw where:
    x = Input
    y=State
    z=Const
    w=DistVar</xs:documentation>
  </xs:annotation>
</xs:element>
element `PosLevel`

```
<xs:element name="PosLevel" default="3">
  <xs:annotation>
    <xs:documentation>Position Level tells the controller how close to come to the programmed location. This is used to allow the robot to round or smooth direction changes. Level 0 makes the robot most accurate. Level 5 makes the smoothest path.</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="0"/>
      <xs:enumeration value="1"/>
      <xs:enumeration value="2"/>
      <xs:enumeration value="3"/>
      <xs:enumeration value="4"/>
      <xs:enumeration value="5"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

```
<xs:element name="RotatingStationPosition" type="RotatingStationJobDataPositionType" minOccurs="0" maxOccurs="4"/>
```
element `ReferenceModel`

Specifies the user frame number or MotoSimEG Cell Model that is used to specify the origin of the part coordinate system.

```xml
<xs:element name="ReferenceModel" default="UF1">
  <xs:annotation>
    <xs:documentation>Specifies the user frame number or MotoSimEG Cell Model that is used to specify the origin of the part coordinate system. </xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value=".{1,15}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

element `RefFrame`

Tells the robot the reference frame of the position. This can be:
- RF for Robot Frame,
- BF for Base Frame,
- TF for tool Frame,
- UFx for User Frame, or it can point to a model in the MotoSmeEG Cell.
<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>element Position</td>
</tr>
<tr>
<td>complexTypes</td>
<td>ConvertType GetCurPos ShiftOn ShiftOnType</td>
</tr>
</tbody>
</table>

| facets              | enumeration BF            |
|                     | enumeration RF            |
|                     | enumeration TF            |
|                     | enumeration UF1           |
|                     | enumeration UF2           |
|                     | enumeration UF3           |
|                     | enumeration UF4           |
|                     | enumeration UF5           |
|                     | enumeration UF6           |
|                     | enumeration UF7           |
|                     | enumeration UF8           |
|                     | enumeration UF9           |
|                     | enumeration UF10          |
|                     | enumeration UF11          |
|                     | enumeration UF12          |
|                     | enumeration UF13          |
|                     | enumeration UF14          |
|                     | enumeration UF15          |
|                     | enumeration UF16          |
|                     | enumeration UF17          |
|                     | enumeration UF18          |
|                     | enumeration UF19          |
|                     | enumeration UF20          |
|                     | enumeration UF21          |
|                     | enumeration UF22          |
|                     | enumeration UF23          |
|                     | enumeration UF24          |
|                     | enumeration MTF           |

<table>
<thead>
<tr>
<th>annotation</th>
<th>documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tells the robot the reference frame of the position. This can be:</td>
<td></td>
</tr>
<tr>
<td>RF for Robot Frame,</td>
<td></td>
</tr>
<tr>
<td>BF for Base Frame,</td>
<td></td>
</tr>
<tr>
<td>TF for tool Frame,</td>
<td></td>
</tr>
<tr>
<td>UFx for User Frame, or it can point to a model in the MotoSmEG Cell.</td>
<td></td>
</tr>
</tbody>
</table>

| source              | <xs:element name="RefFrame"> |
|                     | <xs:annotation> |
|                     | <xs:documentation>Tells the robot the reference frame of the position. This can be: |
|                     | RF for Robot Frame, |
|                     | BF for Base Frame, |
|                     | TF for tool Frame, |
|                     | UFx for User Frame, or it can point to a model in the MotoSmEG Cell.<xs:documentation> |
|                     | </xs:annotation> |
|                     | <xs:simpleType> |
|                     | <xs:restriction base="xs:string"> |
|                     | <xs:enumeration value="BF"/> |
|                     | <xs:enumeration value="RF"/> |
|                     | <xs:enumeration value="TF"/> |
|                     | <xs:enumeration value="UF1"/> |
|                     | <xs:enumeration value="UF2"/> |
|                     | <xs:enumeration value="UF3"/> |
|                     | <xs:enumeration value="UF4"/> |
|                     | <xs:enumeration value="UF5"/> |
|                     | <xs:enumeration value="UF6"/> |
element Root

This is the root element.

source
<xs:element name="Root">
  <xs:annotation>
    <xs:documentation>This is the root element.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Settings"/>
      <xs:element ref="JobData"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
### element Rx

- **Diagram:** ![Rx](image)
- **Type:** `xs:double`
- **Properties:** content `simple`
- **Used by:** elements `Position ThreeDOF`
- **Source:** `&lt;xs:element name="Rx" type="xs:double"/&gt;`

### element Ry

- **Diagram:** ![Ry](image)
- **Type:** `xs:double`
- **Properties:** content `simple`
- **Used by:** elements `Position ThreeDOF`
- **Source:** `&lt;xs:element name="Ry" type="xs:double"/&gt;`

### element Rz

- **Diagram:** ![Rz](image)
- **Type:** `xs:double`
- **Properties:** content `simple`
- **Used by:** elements `Position ThreeDOF`
- **Source:** `&lt;xs:element name="Rz" type="xs:double"/&gt;`

### element S

- **Diagram:** ![S](image)
- **Type:** `xs:long`
- **Properties:** content `simple`
- **Used by:** element `ArmConfig`
- **Source:** `&lt;xs:element name="S" type="xs:long"/&gt;`

### element SetPulseConfig
This command allows you to select the desired robot form or starting configuration. This command causes a PUTP(POS{100X} command in the intermediate JobName.inf file. Where X is the Index Value.
The index is used to select from the ArmConfig commands in the Settings Section of the XML file.

NOTE:
1. Index must be in the range 0 to 11.
element SetPulseConfig/Index

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0 content simple</td>
</tr>
<tr>
<td>facets</td>
<td>totalDigits 2 fractionDigits 0 enumeration 0 enumeration 1 enumeration 2 enumeration 3 enumeration 4 enumeration 5 enumeration 6 enumeration 7 enumeration 8 enumeration 9 enumeration 10 enumeration 11 enumeration 12 enumeration 13 enumeration 14 enumeration 15 enumeration 16 enumeration 17 enumeration 18 enumeration 19 enumeration 20 enumeration 21</td>
</tr>
</tbody>
</table>
**element Settings**

**diagram**

```
<xs:element name="Index">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:totalDigits value="2"/>
      <xs:fractionDigits value="0"/>
      <xs:enumeration value="0"/>
      <xs:enumeration value="1"/>
      <xs:enumeration value="2"/>
      <xs:enumeration value="3"/>
      <xs:enumeration value="4"/>
      <xs:enumeration value="5"/>
      <xs:enumeration value="6"/>
      <xs:enumeration value="7"/>
      <xs:enumeration value="8"/>
      <xs:enumeration value="9"/>
      <xs:enumeration value="10"/>
      <xs:enumeration value="11"/>
      <xs:enumeration value="12"/>
      <xs:enumeration value="13"/>
      <xs:enumeration value="14"/>
      <xs:enumeration value="15"/>
      <xs:enumeration value="16"/>
      <xs:enumeration value="17"/>
      <xs:enumeration value="18"/>
      <xs:enumeration value="19"/>
      <xs:enumeration value="20"/>
      <xs:enumeration value="21"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```
DefaultMotionType

When a position command does not include a MotionType command, the DefaultMotionType is used.

ReferenceModel

Specifies the user frame number or MotoSimEG Cell Model that is used to specify the origin of the part coordinate system.

InputPositionData

ArmConfig

0.21

Specifies the robot initial starting position and form. This position is specified in joint coordinates and will set the initial pose (elbow up/down, etc.) of the robot.

This position can be set by moving the simulated robot to the part coordinate system 0.0.0 location and setting the individual axes to archive the desired pose. The values can be copied from MotoSimEG and pasted in the appropriate joint location.

Note: You can specify up to 21 unique robot configurations.

Note: The first set of Arm Configuration values are automatically selected.

Note: Other Configurations can be selected by the SetPulseConfig commands.

Settings

This is the section of the file that contains the Robot specific and other set up data.

ThreeDOF

Use this values to set the last length of the tool to be perpendicular to the part coordinate system. Generally this will be 180, 0, 0. Where X will be -90, 0, 90, 180, 270 degrees.

ExternalAxes

This element is set to True when the Robot is equipped with an external Axis. Otherwise it is set to False.

Current Converter only supports a static external axis.

AxesType

This element is used with
This is the section of the file that contains the Robot specific and other set up data.

Source:
```xml
<xs:element name="Settings">
  <xs:annotation>
    <xs:documentation>This is the section of the file that contains the Robot specific and other set up data.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Units"/>
      <xs:element ref="AngleSystem"/>
      <xs:element ref="DefaultVelocity"/>
      <xs:element ref="DefaultMotionType"/>
      <xs:choice>
        <xs:element ref="ReferenceModel" minOccurs="0" maxOccurs="21"/>
        <xs:element ref="InputPositionData" minOccurs="0"/>
      </xs:choice>
      <xs:element ref="ArmConfig" minOccurs="0" maxOccurs="21"/>
      <xs:element ref="ThreeDOF" minOccurs="0"/>
      <xs:element ref="ExternalAxes" minOccurs="0"/>
      <xs:element ref="AxesType" minOccurs="0"/>
      <xs:element ref="AxesPulsePosition" minOccurs="0"/>
      <xs:element ref="ToolNumber"/>
      <xs:element ref="JobOutputOptions"/>
      <xs:element ref="BaseStation"/>
      <xs:element ref="RotatingStationSetup"/>
      <xs:element ref="ModelOffset"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```
<xs:element ref="ToolNumber" minOccurs="0"/>
<xs:element ref="JobOutputOptions" minOccurs="0"/>
<xs:element name="BaseStation" type="BaseStationType" minOccurs="0"/>
<xs:element name="RotatingStationSetup" type="RotatingStationSetupType" minOccurs="0" maxOccurs="5"/>
<xs:element name="ModelOffset" type="ModelOffsetType" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>

**element Settings/BaseStation**

**diagram**

```
<xs:element name="BaseStation" type="BaseStationType" minOccurs="0"/>
```

**type** BaseStationType

**properties**
- isRef: 0
- minOccurs: 0
- maxOccurs: 1
- content: complex

**children**
- BSMaximumLength
- BSMaximumCounts
- BSUserFrame
- BSRobot

**source**
```
<xs:element name="BaseStation" type="BaseStationType" minOccurs="0"/>
```

**element Settings/RotatingStationSetup**

**diagram**

```
<xs:element name="RotatingStationSetup" type="RotatingStationSetupType" minOccurs="0"/>
```

**type** RotatingStationSetupType
<table>
<thead>
<tr>
<th>properties</th>
<th>isRef</th>
<th>minOcc</th>
<th>maxOcc</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>complex</td>
</tr>
</tbody>
</table>

| children    | RotationStationName RotatingStationUserFrameNumber PartCordinatesRotateWithStation DefaultRotatingVelocity DefaultRotatingMotionType DefaultRotatingUnits |

| source      | `<xs:element name="RotatingStationSetup" type="RotatingStationSetupType" minOccurs="0" maxOccurs="5"/>` |

**element Settings/ModelOffset**

**diagram**

![Diagram of ModelOffsetType]

<table>
<thead>
<tr>
<th>type</th>
<th>ModelOffsetType</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>properties</th>
<th>isRef</th>
<th>minOcc</th>
<th>maxOcc</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>complex</td>
</tr>
</tbody>
</table>

| children     | ModelOffsetX ModelOffsetY ModelOffsetZ ModelAngleRx ModelAngleRy ModelAngleRz |

| source       | `<xs:element name="ModelOffset" type="ModelOffsetType" minOccurs="0"/>` |

**element SetVar**

**diagram**

![Diagram of SetVar]
SetVar is an XML to JBI Instruction that allows the setting of Robot Variables. These Variables may be B, I, D or R variables.

P variables are set with SetPosElmnt.

This Element may have one of two forms. In form one, you specify the Variable Type and Index and provide the variable value with a "Const" element. In Form 2 you specify the source and destination variable types and indices.

```
xsd:element name="SetVar" type="SetVar">
  <xsd:annotation>
    <xsd:documentation>SetVar is an XML to JBI Instruction that allows the setting of Robot Variables. These Variables may be B, I, D or R variables.

    P variables are set with SetPosElmnt.

    This Element may have one of two forms. In form one, you specify the Variable Type and Index and provide the variable value with a "Const" element. In Form 2 you specify the source and destination variable types and indices.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
```
**element SourceNum**

```xml
<source>
  <xs:element name="SourceNum" type="xs:integer"/>
</source>
```

**element SourceType**

```xml
<source>
  <xs:element name="SourceType" type="xs:string"/>
</source>
```

**element Sub**

```xml
<source>
  <xs:element name="Sub" type="SetVar"/>
</source>
```

**Diagram**

![Diagram](image)
<table>
<thead>
<tr>
<th>diagram</th>
<th>type</th>
<th>properties</th>
<th>used by</th>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>xs:long</td>
<td>content simple</td>
<td>ArmConfig</td>
<td><code>&lt;xs:element name=&quot;T&quot; type=&quot;xs:long&quot;/&gt;</code></td>
</tr>
</tbody>
</table>

**element ThreeDOF**

<table>
<thead>
<tr>
<th>diagram</th>
<th>properties</th>
<th>children</th>
<th>used by</th>
<th>annotation documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>content complex</td>
<td>Rx Ry Rz</td>
<td>Settings</td>
<td>Use this values to set the last length of the tool to be perpendicular to the part coordinate system. Generally this will be 180, 0,X Where X will be -90, 0, 90, 180, 270 degrees.</td>
</tr>
</tbody>
</table>

```xml
<xs:element name="ThreeDOF">
  <xs:annotation>
    <xs:documentation>Use this values to set the last length of the tool to be perpendicular to the part coordinate system. Generally this will be 180, 0,X Where X will be -90, 0, 90, 180, 270 degrees. </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Rx"/>
      <xs:element ref="Ry"/>
      <xs:element ref="Rz"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**element Timer**
### `Timer` Element

- **Type**: `xs:decimal`
- **Properties**: `content simple`
- **Used by**: `JobData`

**Annotation Documentation**

XML to JBI Instruction.

Creates an inform command with the following form:

```
TIMER T=X where X is the number of seconds to delay.
```

**Source**

```xml
<xs:element name="Timer" type="xs:decimal">
  <xs:annotation>
    <xs:documentation>XML to JBI Instruction.
    Creates an inform command with the following form:
    TIMER T=X where X is the number of seconds to delay.</xs:documentation>
  </xs:annotation>
</xs:element>
```

### `TimeUnits` Element

- **Type**: Restriction of `xs:string`
- **Properties**: `content simple`
- **Used by**: `Units`
- **Facets**
  - Enumeration: `Minutes`, `Seconds`

**Source**

```xml
<xs:element name="TimeUnits">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Minutes"/>
      <xs:enumeration value="Seconds"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

### `ToolNumber` Element

- **Type**: Restriction of `xs:int`
- **Properties**: `content simple`
  - **Default**: `0`
used by element Settings

facets
- enumeration 0
- enumeration 1
- enumeration 2
- enumeration 3
- enumeration 4
- enumeration 5
- enumeration 6
- enumeration 7
- enumeration 8
- enumeration 9
- enumeration 10
- enumeration 11
- enumeration 12
- enumeration 13
- enumeration 14
- enumeration 15
- enumeration 16
- enumeration 17
- enumeration 18
- enumeration 19
- enumeration 20
- enumeration 21
- enumeration 22
- enumeration 23
- enumeration 24
- enumeration 25

source
```xml
<xs:element name="ToolNumber" default="0">
  <xs:simpleType>
    <xs:restriction base="xs:int">
      <xs:enumeration value="0"/>
      <xs:enumeration value="1"/>
      <xs:enumeration value="2"/>
      <xs:enumeration value="3"/>
      <xs:enumeration value="4"/>
      <xs:enumeration value="5"/>
      <xs:enumeration value="6"/>
      <xs:enumeration value="7"/>
      <xs:enumeration value="8"/>
      <xs:enumeration value="9"/>
      <xs:enumeration value="10"/>
      <xs:enumeration value="11"/>
      <xs:enumeration value="12"/>
      <xs:enumeration value="13"/>
      <xs:enumeration value="14"/>
      <xs:enumeration value="15"/>
      <xs:enumeration value="16"/>
      <xs:enumeration value="17"/>
      <xs:enumeration value="18"/>
      <xs:enumeration value="19"/>
      <xs:enumeration value="20"/>
      <xs:enumeration value="21"/>
      <xs:enumeration value="22"/>
      <xs:enumeration value="23"/>
      <xs:enumeration value="24"/>
      <xs:enumeration value="25"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```
### element U

<table>
<thead>
<tr>
<th>diagram</th>
<th>![Diagram of element U]</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td><code>xs:long</code></td>
</tr>
<tr>
<td>properties</td>
<td>content <code>simple</code></td>
</tr>
<tr>
<td>used by</td>
<td>element <code>ArmConfig</code></td>
</tr>
<tr>
<td>source</td>
<td><code>&lt;xs:element name=&quot;U&quot; type=&quot;xs:long&quot;/&gt;</code></td>
</tr>
</tbody>
</table>

### element Units

<table>
<thead>
<tr>
<th>diagram</th>
<th>![Diagram of element Units]</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content <code>complex</code></td>
</tr>
<tr>
<td>children</td>
<td><code>LengthUnits AngleUnits TimeUnits</code></td>
</tr>
<tr>
<td>used by</td>
<td>element <code>Settings</code></td>
</tr>
<tr>
<td>annotation documentation</td>
<td>This element, and its sub elements specify the Length, Angle, and Time Units used in this XML File.</td>
</tr>
</tbody>
</table>
| source  | `<xs:element name="Units">  
  <xs:annotation>  
  <xs:documentation>This element, and its sub elements specify the Length, Angle, and Time Units used in this XML File.</xs:documentation>  
  </xs:annotation>  
  <xs:complexType>  
  <xs:sequence>  
  <xs:element ref="LengthUnits"/>  
  <xs:element ref="AngleUnits"/>  
  <xs:element ref="TimeUnits"/>  
  </xs:sequence>  
  </xs:complexType>  
</xs:element>` |

### element X

<table>
<thead>
<tr>
<th>diagram</th>
<th>![Diagram of element X]</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td><code>xs:double</code></td>
</tr>
<tr>
<td>properties</td>
<td>content <code>simple</code></td>
</tr>
</tbody>
</table>
element X

<xs:element name="X" type="xs:double"/>

element Y

<xs:element name="Y" type="xs:double"/>

element Z

<xs:element name="Z" type="xs:double"/>

complexType BaseStationType

<xs:complexType name="BaseStationType">
  <xs:sequence>
    <xs:element name="BSMaximumLength"/>
    <xs:element name="BSMaximumCounts"/>
    <xs:element name="BSUserFrame"/>
    <xs:element name="BSRobot"/>
  </xs:sequence>
</xs:complexType>

used by element Position

source <xs:element name="X" type="xs:double"/>

source <xs:element name="Y" type="xs:double"/>

source <xs:element name="Z" type="xs:double"/>

used by element Settings/BaseStation
<xs:complexType name="BaseStationType">
  <xs:sequence>
    <xs:element name="BSMaximumLength" default="0.0">
      <xs:simpleType>
        <xs:restriction base="xs:float">
          <xs:pattern value="^[+-]?(\d+(\d+)?)$/">
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="BSMaximumCounts" default="0">
      <xs:simpleType>
        <xs:restriction base="xs:int">
          <xs:pattern value="^[+-]?(\d+)$/">
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="BSUserFrame">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="BSUFAxisAlignedWithBase" default="X">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="X"/>
                <xs:enumeration value="Y"/>
                <xs:enumeration value="Z"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="BSUFNumber" default="1">
            <xs:simpleType>
              <xs:restriction base="xs:integer">
                <xs:totalDigits value="2"/>
                <xs:fractionDigits value="0"/>
                <xs:maxInclusive value="25"/>
                <xs:minInclusive value="1"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="BSRobot">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="BSRobotBaseAxis" default="X">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="X"/>
                <xs:enumeration value="Y"/>
                <xs:enumeration value="Z"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="BSRobotMinBaseAxisValue" default="0.0"/>
### element `BaseStationType/BSMaximumLength`

<table>
<thead>
<tr>
<th>Diagram</th>
<th><img src="image" alt="BSMaximumLength" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>restriction of <code>xs:float</code></td>
</tr>
<tr>
<td>Properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>content simple</td>
</tr>
<tr>
<td></td>
<td>default 0.0</td>
</tr>
<tr>
<td>Facets</td>
<td>pattern <code>[+-]?\d+(\.\d+)?</code></td>
</tr>
<tr>
<td>Source</td>
<td><code>&lt;xs:element name=&quot;BSMaximumLength&quot; default=&quot;0.0&quot;&gt;</code></td>
</tr>
</tbody>
</table>

### element `BaseStationType/BSMaximumCounts`

<table>
<thead>
<tr>
<th>Diagram</th>
<th><img src="image" alt="BSMaximumCounts" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>restriction of <code>xs:int</code></td>
</tr>
<tr>
<td>Properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>content simple</td>
</tr>
<tr>
<td></td>
<td>default 0</td>
</tr>
<tr>
<td>Facets</td>
<td>pattern <code>[+-]??\d+</code></td>
</tr>
<tr>
<td>Source</td>
<td><code>&lt;xs:element name=&quot;BSMaximumLength&quot; default=&quot;0.0&quot;&gt;</code></td>
</tr>
</tbody>
</table>
element **BaseStationType/BSUserFrame**

```
<xs:complexType>
  <xs:sequence>
    <xs:element name="BSUFAxisAlignedWithBase" default="X">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="X"/>
          <xs:enumeration value="Y"/>
          <xs:enumeration value="Z"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="BSUFNumber" default="1">
      <xs:simpleType>
        <xs:restriction base="xs:integer">
          <xs:totalDigits value="2"/>
          <xs:fractionDigits value="0"/>
          <xs:maxInclusive value="25"/>
          <xs:minInclusive value="1"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

**Diagram**

```
  BSUserFrame   -----> BSUFAxisAlignedWithBase  -----> BSUFNumber
```

**Properties**
- **isRef**: 0
- **content**: complex

**Children**
- **BSUFAxisAlignedWithBase**
- **BSUFNumber**

**element BaseStationType/BSUserFrame/BSUFAxisAlignedWithBase**

```
<xs:element name="BSUFAxisAlignedWithBase">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="X"/>
      <xs:enumeration value="Y"/>
      <xs:enumeration value="Z"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**Type**
- restriction of **xs:string**
element `BaseStationType/BSUserFrame/BSUFNumber`

diagram

<table>
<thead>
<tr>
<th>properties</th>
<th>isRef 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>content</td>
<td>simple</td>
</tr>
<tr>
<td>default</td>
<td>X</td>
</tr>
</tbody>
</table>

facets

<table>
<thead>
<tr>
<th>enumeration</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>enumeration</td>
<td>Y</td>
</tr>
<tr>
<td>enumeration</td>
<td>Z</td>
</tr>
</tbody>
</table>

source

```xml
<xs:element name="BSUFNumber" default="X">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="X"/>
      <xs:enumeration value="Y"/>
      <xs:enumeration value="Z"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

element `BaseStationType/BSRobot`

diagram

<table>
<thead>
<tr>
<th>properties</th>
<th>isRef 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>content</td>
<td>complex</td>
</tr>
</tbody>
</table>

source

```xml
<xs:element name="BSUFNumber" default="1">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:totalDigits value="2"/>
      <xs:fractionDigits value="0"/>
      <xs:maxInclusive value="25"/>
      <xs:minInclusive value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```
<table>
<thead>
<tr>
<th>children</th>
<th>BSRobotBaseAxis</th>
<th>BSRobotMinBaseAxisValue</th>
<th>BSRobotMaxAxisValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;BSRobot&quot;&gt;</td>
<td><a href="">xs:complexType</a></td>
<td>&lt;xs:element name=&quot;BSRobotBaseAxis&quot; default=&quot;X&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:complexType</a></td>
<td>&lt;xs:element name=&quot;BSRobotBaseAxis&quot; default=&quot;X&quot;&gt;</td>
<td>&lt;xs:restriction base=&quot;xs:string&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:simpleType</a></td>
<td>&lt;xs:restriction base=&quot;xs:string&quot;&gt;</td>
<td>&lt;xs:enumeration value=&quot;&quot;X&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:simpleType</a></td>
<td>&lt;xs:restriction base=&quot;xs:string&quot;&gt;</td>
<td>&lt;xs:enumeration value=&quot;&quot;Y&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:simpleType</a></td>
<td>&lt;xs:restriction base=&quot;xs:string&quot;&gt;</td>
<td>&lt;xs:enumeration value=&quot;&quot;Z&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:restriction&gt;</td>
<td>&lt;/xs:restriction&gt;</td>
<td>&lt;/xs:restriction&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:element&gt;</td>
<td>&lt;/xs:element&gt;</td>
<td>&lt;/xs:element&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element name=&quot;BSRobotMinBaseAxisValue&quot; default=&quot;0.0&quot;&gt;</td>
<td><a href="">xs:complexType</a></td>
<td>&lt;xs:element name=&quot;BSRobotMinBaseAxisValue&quot; default=&quot;0.0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:complexType</a></td>
<td>&lt;xs:element name=&quot;BSRobotMinBaseAxisValue&quot; default=&quot;0.0&quot;&gt;</td>
<td>&lt;xs:restriction base=&quot;xs:float&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:simpleType</a></td>
<td>&lt;xs:restriction base=&quot;xs:float&quot;&gt;</td>
<td>&lt;xs:pattern value=&quot;[+-]?d+(.d+)?&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:simpleType&gt;</td>
<td>&lt;/xs:restriction&gt;</td>
<td>&lt;/xs:restriction&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:element&gt;</td>
<td>&lt;/xs:element&gt;</td>
<td>&lt;/xs:element&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:element name=&quot;BSRobotMaxAxisValue&quot; default=&quot;700.0&quot;&gt;</td>
<td><a href="">xs:complexType</a></td>
<td>&lt;xs:element name=&quot;BSRobotMaxAxisValue&quot; default=&quot;700.0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:complexType</a></td>
<td>&lt;xs:element name=&quot;BSRobotMaxAxisValue&quot; default=&quot;700.0&quot;&gt;</td>
<td>&lt;xs:restriction base=&quot;xs:float&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:simpleType</a></td>
<td>&lt;xs:restriction base=&quot;xs:float&quot;&gt;</td>
<td>&lt;xs:pattern value=&quot;[+-]?d+(.d+)?&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:simpleType&gt;</td>
<td>&lt;/xs:restriction&gt;</td>
<td>&lt;/xs:restriction&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:element&gt;</td>
<td>&lt;/xs:element&gt;</td>
<td>&lt;/xs:element&gt;</td>
</tr>
</tbody>
</table>

**element BaseStationType/BSRobot/BSRobotBaseAxis**

**diagram**

![BSRobotBaseAxis](image)

**type** restriction of xs:string

**properties**

- isRef 0
- content simple
default X

**facets**

- enumeration X
- enumeration Y
- enumeration Z

**source**

```xml
<xs:element name="BSRobotBaseAxis" default="X">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="X"/>
      <xs:enumeration value="Y"/>
      <xs:enumeration value="Z"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```
### element BaseStationType/BSRobot/BSRobotMinBaseAxisValue

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="BSRobotMinBaseAxisValue" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:float</td>
</tr>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>content simple</td>
</tr>
<tr>
<td></td>
<td>default 0.0</td>
</tr>
<tr>
<td>facets</td>
<td>pattern ([-+]?\d+(.\d+))?</td>
</tr>
</tbody>
</table>
| source        | `<xs:element name="BSRobotMinBaseAxisValue" default="0.0">  
|               |   `<xs:restriction base="xs:float">  
|               |     `<xs:pattern value="[+-]?\d+(\.\d+)?"/>  
|               |   `</xs:restriction>`  
|               | `</xs:element>`                  |

### element BaseStationType/BSRobot/BSRobotMaxAxisValue

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="BSRobotMaxAxisValue" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:float</td>
</tr>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>content simple</td>
</tr>
<tr>
<td></td>
<td>default 700.0</td>
</tr>
<tr>
<td>facets</td>
<td>pattern ([-+]?\d+(.\d+))?</td>
</tr>
</tbody>
</table>
| source        | `<xs:element name="BSRobotMaxAxisValue" default="700.0">  
|               |   `<xs:restriction base="xs:float">  
|               |     `<xs:pattern value="[+-]?\d+(\.\d+)?"/>  
|               |   `</xs:restriction>`  
|               | `</xs:element>`                |

### complexType ConvertType
children: **SourceNum**  **DestNum**  **RefFrame**

used by: element **JobData/Convert**

source:
```xml
<xsl:complexType name="ConvertType">
    <xsl:sequence>
        <xsl:element ref="SourceNum"/>
        <xsl:element ref="DestNum"/>
        <xsl:element ref="RefFrame"/>
    </xsl:sequence>
</xsl:complexType>
```

complexType **GetCurPos**

children: **DestNum**  **DataType**  **RefFrame**

used by: element **JobData/GetCurPos**
**complexType GetCurPos**

```xml
<xs:complexType name="GetCurPos">
  <xs:sequence>
    <xs:element ref="DestNum"/>
    <xs:element name="DataType" default="Pulse">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="Pulse"/>
          <xs:enumeration value="Cart"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element ref="RefFrame" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

**element GetCurPos/DataType**

<table>
<thead>
<tr>
<th>Diagram</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td></td>
</tr>
</tbody>
</table>

**type** restriction of `xs:string`

**properties**
- `isRef` 0
- `content` simple
- `default` Pulse

**facets**
- enumeration Pulse
- enumeration Cart

**source**

```xml
<xs:element name="DataType" default="Pulse">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Pulse"/>
      <xs:enumeration value="Cart"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**complexType InputPositionDataType**

<table>
<thead>
<tr>
<th>Diagram</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td></td>
</tr>
</tbody>
</table>

**children**
- InputPositionDataPulse
- InputPositionDataTypeRectan

**used by** element InputPositionData
<xs:complexType name="InputPositionDataType">
    <xs:choice>
        <xs:element name="InputPositionDataPulse" minOccurs="0">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="PositionConfigurationPulse" type="PositionConfigurationPulseType"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="InputPositionDataTypeRectan" minOccurs="0">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="PositionConfiguration" type="PositionConfigurationType"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:choice>
</xs:complexType>

element InputPositionDataType/InputPositionDataPulse

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>isRef 0</td>
</tr>
<tr>
<td>minOcc 0</td>
</tr>
<tr>
<td>maxOcc 1</td>
</tr>
<tr>
<td>content complex</td>
</tr>
</tbody>
</table>

children
- PositionConfigurationPulse

source <xs:element name="InputPositionDataPulse" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="PositionConfigurationPulse" type="PositionConfigurationPulseType"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

element InputPositionDataType/InputPositionDataPulse/PositionConfigurationPulse

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PositionConfigurationPulseType</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>isRef 0</td>
</tr>
<tr>
<td>content complex</td>
</tr>
</tbody>
</table>

children
- BaseFrame
- RobotFrame
**element InputPositionDataType/InputPositionDataTypeRectan**

| properties | isRef 0  
|            | minOcc 0  
|            | maxOcc 1  
| content    | complex  |

**children**: PositionConfiguration

**source**

```xml
<xs:element name="InputPositionDataTypeRectan" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PositionConfiguration" type="PositionConfigurationType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**element InputPositionDataType/InputPositionDataTypeRectan/PositionConfiguration**

| children | RobotFrame BaseFrame UserFrame |

**source**

```xml
<xs:element name="PositionConfiguration" type="PositionConfigurationType"/>
```

**complexType JobOutputOptionsType**

| children | JobType |

**used by**

| element | JobOutputOptions |
<xs:complexType name="JobOutputOptionsType">
  <xs:all>
    <xs:element name="JobType">
      <xs:complexType>
        <xs:choice>
          <xs:element name="PulseJobType" default="True" minOccurs="0">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="True"/>
                <xs:enumeration value="False"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="RectanJobType" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="PositionConfiguration" type="PositionConfigurationType"/>
                <xs:element name="RobotPosture">
                  <xs:complexType>
                    <xs:all>
                      <xs:element name="FlipPosture" default="0">
                        <xs:simpleType>
                          <xs:restriction base="xs:integer">
                            <xs:enumeration value="0"/>
                            <xs:enumeration value="1"/>
                          </xs:restriction>
                        </xs:simpleType>
                      </xs:element>
                      <xs:element name="UpperPosture" default="0">
                        <xs:simpleType>
                          <xs:restriction base="xs:integer">
                            <xs:enumeration value="0"/>
                            <xs:enumeration value="1"/>
                          </xs:restriction>
                        </xs:simpleType>
                      </xs:element>
                      <xs:element name="FrontPosture" default="0">
                        <xs:simpleType>
                          <xs:restriction base="xs:integer">
                            <xs:enumeration value="0"/>
                            <xs:enumeration value="1"/>
                          </xs:restriction>
                        </xs:simpleType>
                      </xs:element>
                      <xs:element name="RPosture" default="0">
                        <xs:simpleType>
                          <xs:restriction base="xs:integer">
                            <xs:enumeration value="0"/>
                            <xs:enumeration value="1"/>
                          </xs:restriction>
                        </xs:simpleType>
                      </xs:element>
                      <xs:element name="TPosture" default="0">
                        <xs:simpleType>
                          <xs:restriction base="xs:integer">
                            <xs:enumeration value="0"/>
                            <xs:enumeration value="1"/>
                          </xs:restriction>
                        </xs:simpleType>
                      </xs:element>
                    </xs:all>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:all>
</xs:complexType>
element JobOutputOptionsType/JobType

Diagram:

```
JobType
  ├── PulseJobType
  └── RectanJobType
```

Properties:
- isRef: 0
- content: complex

Children:
- PulseJobType
- RectanJobType

Source:
```xml
<x:simpleType>
  <xs:restriction base="xs:string">
    <xs:enumeration value="True"/>
    <xs:enumeration value="False"/>
  </xs:restriction>
</xs:simpleType>
```

```
<x:element name="PulseJobType" minOccurs="0">  
  <xs:complexType>
    <xs:choice>
      <xs:element name="PositionConfiguration" type="PositionConfigurationType"/>
      <xs:element name="RobotPosture">  
        <xs:complexType>
          <xs:all>
            <xs:element name="FlipPosture" default="0">  
              <xs:restriction base="xs:integer">  
                <xs:enumeration value="0"/>  
                <xs:enumeration value="1"/>  
              </xs:restriction>
            </xs:element>
          </xs:all>
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

```
<x:element name="RectanJobType" minOccurs="0">  
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PositionConfiguration" type="PositionConfigurationType"/>
      <xs:element name="RobotPosture">  
        <xs:complexType>
          <xs:all>
            <xs:element name="FlipPosture" default="0">  
              <xs:restriction base="xs:integer">  
                <xs:enumeration value="0"/>  
                <xs:enumeration value="1"/>  
              </xs:restriction>
            </xs:element>
          </xs:all>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```
element JobOutputOptionsType/JobType/PulseJobType

<table>
<thead>
<tr>
<th>diagram</th>
<th>PulseJobType</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:string</td>
</tr>
</tbody>
</table>
| properties | isRef 0  
minOcc 0  
maxOcc 1  
content simple  
default True |
| facets | enumeration True  
|        | enumeration False |

**source**

```xml
<xs:element name="PulseJobType" default="True" minOccurs="0">
<xs:simpleType>
    <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
    </xs:restriction>
</xs:simpleType>
</xs:element>
```

**element** `JobOutputOptionsType/JobType/RectanJobType`

**diagram**

![Diagram of `RectanJobType` with `PositionConfiguration` and `RobotPosture`]

**properties**

- isRef 0
- minOccurs 0
- maxOccurs 1
- content complex

**children**

- `PositionConfiguration`
- `RobotPosture`

**source**

```xml
<xs:element name="RectanJobType" minOccurs="0">
<xs:complexType>
    <xs:sequence>
        <xs:element name="PositionConfiguration" type="PositionConfigurationType"/>
        <xs:element name="RobotPosture">
            <xs:complexType>
                <xs:all>
                    <xs:element name="FlipPosture" default="0">
                        <xs:simpleType>
                            <xs:restriction base="xs:integer">
                                <xs:enumeration value="0"/>
                                <xs:enumeration value="1"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:element>
                    <xs:element name="UpperPosture" default="0">
                        <xs:simpleType>
                            <xs:restriction base="xs:integer">
                                <xs:enumeration value="0"/>
                                <xs:enumeration value="1"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:element>
                    <xs:element name="FrontPosture" default="0">
                        <xs:simpleType>
                            <xs:restriction base="xs:integer">
                                <xs:enumeration value="0"/>
                                <xs:enumeration value="1"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:element>
                </xs:all>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
```
element JobOutputOptionsType/JobType/RectanJobType/PositionConfiguration

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="PositionConfiguration diagram" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PositionConfigurationType</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>isRef 0</td>
</tr>
<tr>
<td>content complex</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>children</th>
</tr>
</thead>
<tbody>
<tr>
<td>RobotFrame BaseFrame UserFrame</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;xs:element name=&quot;PositionConfiguration&quot; type=&quot;PositionConfigurationType&quot;/&gt;</code></td>
</tr>
</tbody>
</table>

element JobOutputOptionsType/JobType/RectanJobType/RobotPosture
<xs:element name="RobotPosture">
  <xs:complexType>
    <xs:all>
      <xs:element name="FlipPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="UpperPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="FrontPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="RPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="TPosture" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:all>
  </xs:complexType>
</xs:element>
element JobOutputOptionsType/JobType/RectanJobType/RobotPosture/FlipPosture

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td>content</td>
<td>simple</td>
</tr>
<tr>
<td>default</td>
<td>0</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration 0</td>
</tr>
<tr>
<td></td>
<td>enumeration 1</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;FlipPosture&quot; default=&quot;0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:simpleType</a></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:restriction base=&quot;xs:integer&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;1&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:restriction&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:simpleType&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:element&gt;</td>
</tr>
</tbody>
</table>

element JobOutputOptionsType/JobType/RectanJobType/RobotPosture/UpperPosture

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td>content</td>
<td>simple</td>
</tr>
<tr>
<td>default</td>
<td>0</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration 0</td>
</tr>
<tr>
<td></td>
<td>enumeration 1</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:element name=&quot;UpperPosture&quot; default=&quot;0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td><a href="">xs:simpleType</a></td>
</tr>
<tr>
<td></td>
<td>&lt;xs:restriction base=&quot;xs:integer&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;0&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;xs:enumeration value=&quot;1&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:restriction&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:simpleType&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/xs:element&gt;</td>
</tr>
</tbody>
</table>

element JobOutputOptionsType/JobType/RectanJobType/RobotPosture/FrontPosture
<table>
<thead>
<tr>
<th>Diagram</th>
<th>FrontPosture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>restriction of xs:integer</td>
</tr>
<tr>
<td>Properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>content simple</td>
</tr>
<tr>
<td></td>
<td>default 0</td>
</tr>
<tr>
<td>Facets</td>
<td>enumeration 0</td>
</tr>
<tr>
<td></td>
<td>enumeration 1</td>
</tr>
<tr>
<td>Source</td>
<td><code>&lt;xs:element name=&quot;FrontPosture&quot; default=&quot;0&quot;&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:simpleType&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:restriction base=&quot;xs:integer&quot;&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;0&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;1&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/xs:restriction&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/xs:simpleType&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/xs:element&gt;</code></td>
</tr>
</tbody>
</table>

**element** JobOutputOptionsType/JobType/RectanJobType/RobotPosture/RPosture

<table>
<thead>
<tr>
<th>Diagram</th>
<th>RPosture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>restriction of xs:integer</td>
</tr>
<tr>
<td>Properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>content simple</td>
</tr>
<tr>
<td></td>
<td>default 0</td>
</tr>
<tr>
<td>Facets</td>
<td>enumeration 0</td>
</tr>
<tr>
<td></td>
<td>enumeration 1</td>
</tr>
<tr>
<td>Source</td>
<td><code>&lt;xs:element name=&quot;RPosture&quot; default=&quot;0&quot;&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:simpleType&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:restriction base=&quot;xs:integer&quot;&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;0&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;xs:enumeration value=&quot;1&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/xs:restriction&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/xs:simpleType&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/xs:element&gt;</code></td>
</tr>
</tbody>
</table>

**element** JobOutputOptionsType/JobType/RectanJobType/RobotPosture/TPosture

<table>
<thead>
<tr>
<th>Diagram</th>
<th>TPosture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>restriction of xs:integer</td>
</tr>
<tr>
<td>Properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>content simple</td>
</tr>
<tr>
<td></td>
<td>default 0</td>
</tr>
<tr>
<td>Facets</td>
<td>enumeration 0</td>
</tr>
<tr>
<td></td>
<td>enumeration 1</td>
</tr>
</tbody>
</table>
complexType ModelOffsetType

diagram

children
- ModelOffsetX
- ModelOffsetY
- ModelOffsetZ
- ModelAngleRx
- ModelAngleRy
- ModelAngleRz

used by
- element Settings/ModelOffset

source
<xs:complexType name="ModelOffsetType">
  <xs:sequence>
    <xs:element name="ModelOffsetX" type="xs:double" default="0.0" minOccurs="0" maxOccurs="1"/>
    <xs:element name="ModelOffsetY" type="xs:double" default="0.0" minOccurs="0" maxOccurs="1"/>
    <xs:element name="ModelOffsetZ" type="xs:double" default="0.0" minOccurs="0" maxOccurs="1"/>
    <xs:element name="ModelAngleRx" type="xs:double" default="0.0" minOccurs="0" maxOccurs="1"/>
    <xs:element name="ModelAngleRy" type="xs:double" default="0.0" minOccurs="0" maxOccurs="1"/>
    <xs:element name="ModelAngleRz" type="xs:double" default="0.0" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

element ModelOffsetType/ModelOffsetX

diagram

properties
- isRef 0
- minOccurs 0
- maxOccurs 1
- content simple
- default 0.0

source
<xs:element name="ModelOffsetX" type="xs:double" default="0.0" minOccurs="0" maxOccurs="1"/>

element ModelOffsetType/ModelOffsetY

diagram

properties
- isRef 0
- minOccurs 0
- maxOccurs 1
- content simple
- default 0.0

source
<xs:element name="ModelOffsetY" type="xs:double" default="0.0" minOccurs="0" maxOccurs="1"/>
<table>
<thead>
<tr>
<th>diagram</th>
<th>type</th>
<th>xs:double</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td><code>&lt;xs:element name=&quot;ModelOffsetY&quot; type=&quot;xs:double&quot; default=&quot;0.0&quot; minOccurs=&quot;0&quot;/&gt;</code></td>
<td></td>
</tr>
</tbody>
</table>

**Element ModelOffsetType/ModelOffsetZ**

<table>
<thead>
<tr>
<th>diagram</th>
<th>type</th>
<th>xs:double</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td><code>&lt;xs:element name=&quot;ModelOffsetZ&quot; type=&quot;xs:double&quot; default=&quot;0.0&quot; minOccurs=&quot;0&quot;/&gt;</code></td>
<td></td>
</tr>
</tbody>
</table>

**Element ModelOffsetType/ModelAngleRx**

<table>
<thead>
<tr>
<th>diagram</th>
<th>type</th>
<th>xs:double</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td><code>&lt;xs:element name=&quot;ModelAngleRx&quot; type=&quot;xs:double&quot; default=&quot;0.0&quot; minOccurs=&quot;0&quot;/&gt;</code></td>
<td></td>
</tr>
</tbody>
</table>

**Element ModelOffsetType/ModelAngleRy**

<table>
<thead>
<tr>
<th>diagram</th>
<th>type</th>
<th>xs:double</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td><code>&lt;xs:element name=&quot;ModelAngleRy&quot; type=&quot;xs:double&quot; default=&quot;0.0&quot; minOccurs=&quot;0&quot;/&gt;</code></td>
<td></td>
</tr>
</tbody>
</table>

**Element ModelOffsetType/ModelAngleRz**

<table>
<thead>
<tr>
<th>diagram</th>
<th>type</th>
<th>xs:double</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td><code>&lt;xs:element name=&quot;ModelAngleRz&quot; type=&quot;xs:double&quot; default=&quot;0.0&quot; minOccurs=&quot;0&quot;/&gt;</code></td>
<td></td>
</tr>
</tbody>
</table>
complexType PositionConfigurationType

<xs:complexType name="PositionConfigurationType">
  <xs:choice>
    <xs:element name="RobotFrame" type="xs:string" default="True" minOccurs="0"/>
    <xs:element name="BaseFrame" default="False" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
          <xs:enumeration value="False"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="UserFrame" default="1" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:integer">
          <xs:totalDigits value="2"/>
          <xs:fractionDigits value="0"/>
          <xs:minInclusive value="0"/>
          <xs:maxInclusive value="25"/>
          <xs:pattern value="\d{1,2}"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:choice>
</xs:complexType>
element PositionConfigurationType/RobotFrame

type xs:string

properties
  isRef 0
  minOcc 0
  maxOcc 1
  content simple
  default True

source
  <xs:element name="RobotFrame" type="xs:string" default="True" minOccurs="0"/>


element PositionConfigurationType/BaseFrame

type restriction of xs:string

properties
  isRef 0
  minOcc 0
  maxOcc 1
  content simple
  default False

facets
  enumeration True
  enumeration False

source
  <xs:element name="BaseFrame" default="False" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="True"/>
        <xs:enumeration value="False"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>


element PositionConfigurationType/UserFrame

type restriction of xs:integer

properties
  isRef 0
  minOcc 0
  maxOcc 1
  content simple
  default 1

facets
  minInclusive 0
  maxInclusive 25
  totalDigits 2
  fractionDigits 0
  pattern \d{1,2}
<xs:element name="UserFrame" default="1" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:totalDigits value="2"/>
      <xs:fractionDigits value="0"/>
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="25"/>
      <xs:pattern value="\d{1,2}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

complexType PositionConfigurationPulseType

diagram

children
  BaseFrame RobotFrame

used by element InputPositionDataType/InputPositionDataPulse/PositionConfigurationPulse

source <xs:complexType name="PositionConfigurationPulseType">
  <xs:sequence>
    <xs:element name="BaseFrame" default="False" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="True"/>
          <xs:enumeration value="False"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="RobotFrame" type="xs:string" default="True" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

element PositionConfigurationPulseType/BaseFrame

diagram

type restriction of xs:string

properties
  isRef 0
  minOccurs 0
  maxOccurs 1
  content simple
  default False

facets
  enumeration True
  enumeration False
element PositionConfigurationPulseType/RobotFrame

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
</table>

properties

<table>
<thead>
<tr>
<th>isRef</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>minOccurs</td>
<td>0</td>
</tr>
<tr>
<td>maxOcc</td>
<td>1</td>
</tr>
<tr>
<td>content</td>
<td>simple</td>
</tr>
<tr>
<td>default</td>
<td>True</td>
</tr>
</tbody>
</table>

source

<xs:element name="RobotFrame" type="xs:string" default="True" minOccurs="0"/>

complexType RotatingStationJobDataPositionType

diagram

children

RotationStationName PositonUnits PositionValue RotaingVelocity RotatingMotionType

used by element Position/RotatingStationPosition

source

<xs:complexType name="RotatingStationJobDataPositionType">
  <xs:sequence>
    <xs:element name="RotationStationName" default="ST1">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:pattern value="ST1"/>
          <xs:pattern value="ST2"/>
          <xs:pattern value="ST3"/>
          <xs:pattern value="ST4"/>
          <xs:pattern value="ST5"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="PositonUnits" default="Degrees" minOccurs="0">
      <xs:simpleType>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```xml
<xs:element name="RotationStationName">
  
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="Pulse"/>
      <xs:pattern value="Degrees"/>
      <xs:pattern value="Radians"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:element name="PositionValue" default="0.0">
    <xs:simpleType>
      <xs:restriction base="xs:double">
        <xs:minInclusive value="-10000.0"/>
        <xs:maxInclusive value="10000.0"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>

  <xs:element name="RotaingVelocity" default="25" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="100"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>

  <xs:element name="RotatingMotionType" default="Joint" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:pattern value="Joint"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>

</xs:element>
```

---

<table>
<thead>
<tr>
<th>diagram</th>
<th>RotationStationName</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:string</td>
</tr>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td>content</td>
<td>simple</td>
</tr>
<tr>
<td>default</td>
<td>ST1</td>
</tr>
<tr>
<td>facets</td>
<td>ST1, ST2, ST3, ST4, ST5</td>
</tr>
</tbody>
</table>
element RotatingStationJobDataPositionType/PositionUnits

diagram

type restriction of xs:string

properties
  isRef 0
  minOccurs 0
  maxOccurs 1
  content simple
  default Degrees

facets
  pattern Pulse
  pattern Degrees
  pattern Radians

source <xs:element name="PositionUnits" default="Degrees" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="Pulse"/>
      <xs:pattern value="Degrees"/>
      <xs:pattern value="Radians"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

element RotatingStationJobDataPositionType/PositionValue

diagram

type restriction of xs:double

properties
  isRef 0
  content simple
  default 0.0

facets
  minInclusive -10000.0
  maxInclusive 10000.0
**element RotatingStationJobDataPositionType/RotatingVelocity**

- **type**: restriction of `xs:integer`
- **properties**:
  - `isRef`: 0
  - `minOcc`: 0
  - `maxOcc`: 1
  - `content`: simple
  - `default`: 25

- **facets**:
  - `minInclusive`: 1
  - `maxInclusive`: 100

**source**

```xml
<xs:element name="RotatingVelocity" default="25" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="100"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**element RotatingStationJobDataPositionType/RotatingMotionType**

- **type**: restriction of `xs:string`
- **properties**:
  - `isRef`: 0
  - `minOcc`: 0
  - `maxOcc`: 1
  - `content`: simple
  - `default`: Joint

- **facets**: `pattern` Joint

**source**

```xml
<xs:element name="RotatingMotionType" default="Joint" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="Joint"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**complexType RotatingStationPositionType**
source

```xml
<xs:complexType name="RotatingStationPositionType">
  <xs:choice>
    <xs:element name="RotationPositionCartDegrees"/>
    <xs:element name="RotatingPositionPulseCounts"/>
  </xs:choice>
</xs:complexType>
```

---

**element RotatingStationPositionType/RotationPositionCartDegrees**

source

```xml
<xs:element name="RotationPositionCartDegrees"/>
```

---

**element RotatingStationPositionType/RotatingPositionPulseCounts**

source

```xml
<xs:element name="RotatingPositionPulseCounts"/>
```

---

**complexType RotatingStationSetupType**

children

- RotationStationName
- RotatingStationUserFrameNumber
- PartCordinatesRotateWithStation
- DefaultRotatingVelocity
- DefaultRotatingMotionType
- DefaultRotatingUnits

used by

- element Settings/RotatingStationSetup
<xs:complexType name="RotatingStationSetupType">
    <xs:sequence>
        <xs:element name="RotationStationName" default="ST1">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="ST1"/>
                    <xs:enumeration value="ST2"/>
                    <xs:enumeration value="ST3"/>
                    <xs:enumeration value="ST4"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="RotatingStationUserFrameNumber" default="1">
            <xs:simpleType>
                <xs:restriction base="xs:integer">
                    <xs:totalDigits value="2"/>
                    <xs:fractionDigits value="0"/>
                    <xs:minInclusive value="1"/>
                    <xs:maxInclusive value="24"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="PartCordinatesRotateWithStation" default="True">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:pattern value="True"/>
                    <xs:pattern value="False"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="DefaultRotatingVelocity" default="25">
            <xs:simpleType>
                <xs:restriction base="xs:integer">
                    <xs:minInclusive value="0"/>
                    <xs:maxInclusive value="100"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="DefaultRotatingMotionType" default="Joint">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:pattern value="Joint"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="DefaultRotatingUnits" default="Degrees">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:pattern value="EncoderCounts"/>
                    <xs:pattern value="Degrees"/>
                    <xs:pattern value="Radians"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
element RotatingStationSetupType/RotationStationName

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| properties | isRef 0 |
|            | content simple |
|            | default ST1 |

| facets | enumeration ST1 |
|        | enumeration ST2 |
|        | enumeration ST3 |
|        | enumeration ST4 |

| source | <xs:element name="RotationStationName" default="ST1"> |
|        | <xs:simpleType> |
|        | <xs:restriction base="xs:string"> |
|        | <xs:enumeration value="ST1"/> |
|        | <xs:enumeration value="ST2"/> |
|        | <xs:enumeration value="ST3"/> |
|        | <xs:enumeration value="ST4"/> |
|        | </xs:restriction> |
|        | </xs:simpleType> |
|        | </xs:element> |

element RotatingStationSetupType/RotatingStationUserFrameNumber

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:integer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| properties | isRef 0 |
|            | content simple |
|            | default 1 |

| facets | minInclusive 1 |
|        | maxInclusive 24 |
|        | totalDigits 2 |
|        | fractionDigits 0 |

| source | <xs:element name="RotatingStationUserFrameNumber" default="1"> |
|        | <xs:simpleType> |
|        | <xs:restriction base="xs:integer"> |
|        | <xs:totalDigits value="2"/> |
|        | <xs:fractionDigits value="0"/> |
|        | <xs:minInclusive value="1"/> |
|        | <xs:maxInclusive value="24"/> |
|        | </xs:restriction> |
|        | </xs:simpleType> |
|        | </xs:element> |

element RotatingStationSetupType/PartCordinatesRotateWithStation
### Type restriction of `xs:string`

<table>
<thead>
<tr>
<th>diagram</th>
<th>PartCordinatesRotateWithStation</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of <code>xs:string</code></td>
</tr>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td>content</td>
<td>simple</td>
</tr>
<tr>
<td>default</td>
<td>True</td>
</tr>
<tr>
<td>facets</td>
<td>pattern True</td>
</tr>
<tr>
<td>pattern</td>
<td>False</td>
</tr>
</tbody>
</table>

#### Source

```xml
diagram
<xs:element name="PartCordinatesRotateWithStation" default="True">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="True"/>
      <xs:pattern value="False"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

### Element `RotatingStationSetupType/DefaultRotatingVelocity`

<table>
<thead>
<tr>
<th>diagram</th>
<th>DefaultRotatingVelocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of <code>xs:integer</code></td>
</tr>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td>content</td>
<td>simple</td>
</tr>
<tr>
<td>default</td>
<td>25</td>
</tr>
<tr>
<td>facets</td>
<td>minInclusive 0</td>
</tr>
<tr>
<td>maxInclusive</td>
<td>100</td>
</tr>
</tbody>
</table>

#### Source

```xml
diagram
<xs:element name="DefaultRotatingVelocity" default="25">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="100"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

### Element `RotatingStationSetupType/DefaultRotatingMotionType`

<table>
<thead>
<tr>
<th>diagram</th>
<th>DefaultRotatingMotionType</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of <code>xs:string</code></td>
</tr>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td>content</td>
<td>simple</td>
</tr>
<tr>
<td>default</td>
<td>Joint</td>
</tr>
<tr>
<td>facets</td>
<td>pattern Joint</td>
</tr>
</tbody>
</table>

#### Source

```xml
diagram
<xs:element name="DefaultRotatingMotionType" default="Joint">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="Joint"/>
      <xs:pattern value=""/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```
element RotatingStationSetupType/DefaultRotatingUnits

complexType Search

children DistVar State Input Const

used by element Position/Search
element Search/State

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:string</th>
</tr>
</thead>
</table>
| properties | isRef 0  
             | content simple 
             | default Off |
| facets   | enumeration On 
             | enumeration Off |
| source   | <xs:element name="State" default="Off"> 
             | <xs:simpleType> 
             | <xs:restriction base="xs:string"> 
             | <xs:enumeration value="On"/> 
             | <xs:enumeration value="Off"/> 
             | </xs:restriction> 
             | </xs:simpleType> 
             | </xs:element> |

element Search/Input

diagram

<table>
<thead>
<tr>
<th>type</th>
<th>restriction of xs:integer</th>
</tr>
</thead>
</table>
| properties | isRef 0  
             | content simple 
             | default 1 |
| source   | <xs:element name="Input" default="1"> 
             | <xs:simpleType> 
             | <xs:restriction base="xs:integer"> 
             | <xs:totalDigits value="3"/> 
             | <xs:fractionDigits value="0"/> 
             | </xs:restriction> 
             | </xs:simpleType> 
<pre><code>         | &lt;/xs:element&gt; |
</code></pre>
<table>
<thead>
<tr>
<th>facets</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>totalDigits</td>
<td>3</td>
</tr>
<tr>
<td>fractionDigits</td>
<td>0</td>
</tr>
</tbody>
</table>

**source**

```xml
<xs:element name="Input" default="1">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:totalDigits value="3"/>
      <xs:fractionDigits value="0"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**element Search/Const**

**diagram**

![Const](image)

**type** `xs:short`

**properties**

- **isRef**: 0
- **content**: simple
- **default**: 0

**used by** complexType `SetVar`

**source**

```xml
<xs:element name="Const" type="xs:short" default="0"/>
```

**complexType SetPosElmnt**

**diagram**

![SetPosElmnt](image)

**children**

- `SourceNum`
- `DestNum`
- `DestIndex`

**used by** element `JobData/SetPosElmnt`

**source**

```xml
<xs:complexType name="SetPosElmnt">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="SourceNum"/>
    <xs:element ref="DestNum"/>
    <xs:element ref="DestIndex"/>
  </xs:sequence>
</xs:complexType>
```

**complexType SetVar**
children: Const DestType DestIndex SourceNum SourceType DestNum DestType

used by: Add Divide SetVar Sub

source:
<xs:complexType name="SetVar">
  <xs:choice minOccurs="0">
    <xs:sequence>
      <xs:element ref="Const"/>
      <xs:element name="DestType" type="xs:anyType"/>
    </xs:sequence>
    <xs:sequence>
      <xs:element ref="DestIndex"/>
    </xs:sequence>
    <xs:sequence>
      <xs:element ref="SourceNum"/>
      <xs:element ref="SourceType"/>
      <xs:element name="DestNum" type="xs:anyType"/>
      <xs:element ref="DestType"/>
    </xs:sequence>
  </xs:choice>
</xs:complexType>

element SetVar/DestType

type: xs:anyType

properties:
- isRef: 0
- content: complex
- mixed: true
element **SetVar/DestNum**

```xml
<xs:element name="DestType" type="xs:anyType"/>
```

**complexType** **ShiftOn**

```xml
<xs:complexType name="ShiftOn">
    <xs:sequence>
        <xs:element ref="SourceNum"/>
        <xs:element ref="RefFrame"/>
    </xs:sequence>
</xs:complexType>
```

**used by** complexTypes **ConvertType GetCurPos SetPosElmnt**

**children** **SourceNum RefFrame**

**used by** element **JobData/Shft**

**source**

```xml
<xs:complexType name="ShiftOn">
    <xs:sequence>
        <xs:element ref="SourceNum"/>
        <xs:element ref="RefFrame"/>
    </xs:sequence>
</xs:complexType>
```
**complexType** ShiftOnType

### Diagram
![Diagram](https://via.placeholder.com/150)

- **SourceNum**
- **RefFrame**

### Children
- **SourceNum**
- **RefFrame**

### Used by
- **element** JobData/ShiftOn

### Source
```xml
<xs:complexType name="ShiftOnType">
  <xs:sequence>
    <xs:element ref="SourceNum"/>
    <xs:element ref="RefFrame"/>
  </xs:sequence>
</xs:complexType>
```

Tells the robot the reference frame of the position. This can be:
- RF for Robot Frame,
- BF for Base Frame,
- TF for tool Frame,
- UFX for User Frame, or it can point to a model in the MotoSmEG Cell.
Schema **PointstoJBIConverterSchemaV1.xsd**

**schema location:**  
C:\Program Files\Motoman\MotoSim EG\PointsImporterEGSchemas\PointstoJBIConverterSchemaV1.xsd

**attribute form default:**  
**element form default:**  

**Elements**

- AutoReadLastConfigFiles
- AutoReadLastPointsFile
- InformFileFormats
- RegExParseInt
- Root
- Settings

**element AutoReadLastConfigFiles**

<table>
<thead>
<tr>
<th>Diagram</th>
<th>AutoReadLastConfigFiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>restriction of xs:string</td>
</tr>
</tbody>
</table>
| Properties | content simple  
default True |
| Used by | element Settings |
| Facets | enumeration True  
enumeration False |
| Source | <xs:element name="AutoReadLastConfigFiles" default="True">  
<xs:simpleType>  
<xs:restriction base="xs:string">  
<xs:enumeration value="True"/>  
<xs:enumeration value="False"/>  
</xs:restriction>  
</xs:simpleType>  
</xs:element> |

**element AutoReadLastPointsFile**

<table>
<thead>
<tr>
<th>Diagram</th>
<th>AutoReadLastPointsFile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>restriction of xs:string</td>
</tr>
</tbody>
</table>
| Properties | content simple  
default True |
| Used by | element Settings |
| Source | <xs:element name="AutoReadLastPointsFile" default="True">  
<xs:simpleType>  
<xs:restriction base="xs:string">  
<xs:enumeration value="True"/>  
<xs:enumeration value="False"/>  
</xs:restriction>  
</xs:simpleType>  
</xs:element> |
| facets | enumeration False  
|        | enumeration True  |
| source | `<xs:element name="AutoReadLastPointsFile" default="True">  
|        | `<xs:simpleType>  
|        | `<xs:restriction base="xs:string">  
|        | `<xs:enumeration value="False"/>  
|        | `<xs:enumeration value="True"/>  
|        | `</xs:restriction>`  
|        | `</xs:simpleType>`  
|        | `</xs:element>` |

**element InformFileFormats**

| diagram | ![InformFileFormats Diagram](image)

| properties | content complex |

| attributes | Name | Type | Use | Default | Fixed | annotation |
|            | MaxLines | derived | required | by: xs: | short |

| source | `<xs:element name="InformFileFormats">  
|        | `<xs:complexType>  
|        | `<xs:attribute name="MaxLines" use="required">  
|        | `<xs:simpleType>  
|        | `<xs:restriction base="xs:short">  
|        | `<xs:minInclusive value="100"/>  
|        | `<xs:maxInclusive value="1000"/>  
|        | `</xs:restriction>`  
|        | `</xs:complexType>`  
|        | `</xs:attribute>`  
|        | `</xs:complexType>`  
|        | `</xs:element>` |

**attribute InformFileFormats/@MaxLines**

| type | restriction of xs:short |
| properties | isRef 0  
|            | use required |
| facets | minInclusive 100  
|        | maxInclusive 1000 |
element `RegExParseInt`

**diagram**

![RegExParseInt diagram]

**type** `xs:string`

**properties**
- content simple
- default `([\W]+\-[+]*\d{1,3}\d{0,4}\s\d{0,5})*`  

**used by**
- element `Settings`

**source**

```xml
<xs:element name="RegExParseInt" type="xs:string" default="([\W]+\-[+]*\d{1,3}\d{0,4}\s\d{0,5})*
```  

---

**element Root**

**diagram**

![Root diagram]

**properties**
- content complex

**children**
- `Settings`

**source**

```xml
<xs:element name="Root">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="Settings"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```  

---

**element Settings**
## Diagram

![Diagram of XML elements]

<table>
<thead>
<tr>
<th>Properties</th>
<th>Content: complex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Children</td>
<td><code>RegExParseInt</code> <code>AutoReadLastPointsFile</code> <code>AutoReadLastConfigFiles</code></td>
</tr>
<tr>
<td>Used by</td>
<td><code>Root</code> element</td>
</tr>
</tbody>
</table>

```xml
<xs:element name="Settings">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="RegExParseInt" maxOccurs="unbounded"/>
      <xs:element ref="AutoReadLastPointsFile"/>
      <xs:element ref="AutoReadLastConfigFiles"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```
### Elements Complex types

<table>
<thead>
<tr>
<th>B</th>
<th>ArmConfigType</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td></td>
</tr>
<tr>
<td>Root</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td></td>
</tr>
<tr>
<td>Settings</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td></td>
</tr>
<tr>
<td>U</td>
<td></td>
</tr>
</tbody>
</table>

#### element B

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="B Diagram" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:short</td>
</tr>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>complexType ArmConfigType</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration 26546</td>
</tr>
</tbody>
</table>
| source | <xs:element name="B">
  <xs:simpleType>
    <xs:restriction base="xs:short">
      <xs:enumeration value="26546"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element> |

#### element L

<table>
<thead>
<tr>
<th>diagram</th>
<th><img src="image" alt="L Diagram" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:short</td>
</tr>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>complexType ArmConfigType</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration -16903</td>
</tr>
</tbody>
</table>
element R

<table>
<thead>
<tr>
<th>diagram</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restriction of xs:byte</td>
</tr>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>used by</td>
<td>complexType ArmConfigType</td>
</tr>
<tr>
<td>facets</td>
<td>enumeration 0</td>
</tr>
</tbody>
</table>
| source   | <xs:element name="R">  
                 <xs:simpleType>  
                     <xs:restriction base="xs:byte">  
                         <xs:enumeration value="0"/>  
                     </xs:restriction>  
                 </xs:simpleType>  
             </xs:element> |

element Root

<table>
<thead>
<tr>
<th>diagram</th>
<th>Root ➔ Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>content complex</td>
</tr>
<tr>
<td>children</td>
<td>Settings</td>
</tr>
</tbody>
</table>
| source   | <xs:element name="Root">  
                 <xs:complexType>  
                     <xs:sequence>  
                         <xs:element ref="Settings"/>  
                     </xs:sequence>  
                 </xs:complexType>  
             </xs:element> |

element S
### Type restriction of xs:short

- **Properties**: content simple
- **Used by**: complexType `ArmConfigType`
- **Facets**: enumeration 24728

#### XML Source
```
<xs:element name="S">
  <xs:simpleType>
    <xs:restriction base="xs:short">
      <xs:enumeration value="24728"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

### Element Settings

- **Diagram**: ![Diagram](image)
- **Properties**: content complex
- **Children**: `ArmConfig`
- **Used by**: element `Root`

#### XML Source
```
<xs:element name="Settings">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ArmConfig" type="ArmConfigType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

### Element Settings/ArmConfig
**ArmConfigType**

**IsRef:** 0

**Content:** complex

**Children:** S L U R B T

**Attributes:**
- **Name:** JobName
- **Type:** xs:string
- **Use:** optional

**Source:**
```
<xs:element name="ArmConfig" type="ArmConfigType"/>
```

**Element T**

**Diagram:**
```
T
```

**Type:** restriction of xs:int

**Properties:**
- content simple

**Used by:** complexType ArmConfigType

**Facets:**
- enumeration -56400
**element U**

- **Diagram**
  ![Diagram of element U]

- **Type**
  restriction of `xs:int`

- **Properties**
  content: simple

- **Used by**
  complexType: ArmConfigType

- **Facets**
  enumeration: -52854

**source**

```xml
<xs:element name="U">
    <xs:simpleType>
        <xs:restriction base="xs:int">
            <xs:enumeration value="-52854"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

**complexType ArmConfigType**

**Diagram**

![Diagram of complexType ArmConfigType]
### ArmConfigType

Specifies the robot initial starting position and form. This position is specified in joint coordinates and will set the initial pose (elbow up/down, etc.) of the robot.

This position can be set by moving the simulated robot to the part coordinate system 0,0,0 location and setting the individual axes to achieve the desired pose. The values can be copied from MotoSimEG and pasted in the appropriate joint location.

Note: You can specify up to 21 unique robot configurations.

Note: The first set of Arm Configuration values are automatically selected.

<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JobName</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### annotation
documentation

Specifies the robot initial starting position and form. This position is specified in joint coordinates and will set the initial pose (elbow up/down, etc.) of the robot.

This position can be set by moving the simulated robot to the part coordinate system 0,0,0 location and setting the individual axes to achieve the desired pose. The values can be copied from MotoSimEG and pasted in the appropriate joint location.

Note: You can specify up to 21 unique robot configurations.

Note: The first set of Arm Configuration values are automatically selected.
Specifies the robot initial starting position and form. This position is specified in joint coordinates and will set the initial pose (elbow up/dow, etc.) of the robot.

This position can be set by moving the simulated robot to the part coordinate system 0,0,0 location and setting the individual axes to archive the desired pose. The values can be copied from MotoSimEG and pasted in the appropriate joint location.

Note: You can specify up to 21 unique robot configurations.

Note: The first set of Arm Configuration values are automatically selected.

attribute **ArmConfigType/@JobName**

<table>
<thead>
<tr>
<th>type</th>
<th>xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>isRef 0</td>
</tr>
<tr>
<td></td>
<td>use optional</td>
</tr>
</tbody>
</table>

source `<xs:attribute name="JobName" type="xs:string" use="optional"/>`
Index

A
About This Document , 3
AlignXwithPath , 58
Anchor Requirements , 121 , 125
Anchor Requirements and Hardware , 121 , 125
AngleSystem , 57
ANSI/RIA , 13
ArmConfig , 58

B
Basic Introduction , 121
Batch Job Processing , 22

C
Calibration , 126
Calibration Tasks For Offline Filter Creation , 125
CallJob , 63
CircleFixUp , 58
CircleIJCenter , 59
CNC to Robot Orient. Mapping , 51
Config Control Panel , 50
Creating a Job , 22
Customer Service , 12

D
DCISimpleDrip , 59
Debug Options , 51
Default Joint and Linear Velocities , 51
Default Orientation , 51
DefaultNCPath , 59
DefaultToolNumber , 59
Description , 60

E
EndCannedCycle , 61
ExternalAxes , 59

F
First Time Use , 10
FourToSixAxis , 57

G
GCODE , 60
GenInform , 63
GTOINFORM , 59

H
Hardware Key , 10
Header , 60

I
INFORM , 61
InformFileFormats , 56
Installation Safety , 15
Installing G-Code Converter , 42 , 95
Introduction , 1
isLiteralInform , 61

J
JBI Log Panel , 49
Job Conversion Reports , 93

L
LengthUnits , 62
Log File , 92

M
Maintenance Safety , 17
MotionFunction , 60
MotionFunctions , 57
MotionGroup , 61
MotionType , 62

N
NC File Panel , 46
NC to XML Instructions Panel , 47

O
Operation Safety , 17
OutputOn , 62

P
ParseGroups , 56
ParseGroupsThrowAwayLines , 56
Part Programming Tasks , 126
Programming Safety , 16
<table>
<thead>
<tr>
<th>Reference Model</th>
<th>58, 63</th>
</tr>
</thead>
<tbody>
<tr>
<td>RegExParseInt</td>
<td>55</td>
</tr>
<tr>
<td>Robot JBI Files Panel</td>
<td>49</td>
</tr>
<tr>
<td>Robot XML Files Panel</td>
<td>48</td>
</tr>
<tr>
<td>Robotic Industries Association</td>
<td>13</td>
</tr>
<tr>
<td>Safeguarding Tips</td>
<td>15</td>
</tr>
<tr>
<td>Safety</td>
<td>13</td>
</tr>
<tr>
<td>Safety Devices</td>
<td>15</td>
</tr>
<tr>
<td>Sample Files</td>
<td>42, 96</td>
</tr>
<tr>
<td>SetPosElmnt</td>
<td>63</td>
</tr>
<tr>
<td>ShiftOffset</td>
<td>61</td>
</tr>
<tr>
<td>Simulation Tasks</td>
<td>127</td>
</tr>
<tr>
<td>StartCannedCycle</td>
<td>61</td>
</tr>
<tr>
<td>StartFile</td>
<td>56</td>
</tr>
<tr>
<td>Summary Panel</td>
<td>50</td>
</tr>
<tr>
<td>System Requirements</td>
<td>8</td>
</tr>
<tr>
<td>ThreeDOF</td>
<td>58, 62</td>
</tr>
<tr>
<td>ThrowAwayLines</td>
<td>56</td>
</tr>
<tr>
<td>ThrowAwayLinesRegEx</td>
<td>56</td>
</tr>
<tr>
<td>Timer</td>
<td>62</td>
</tr>
<tr>
<td>Units</td>
<td>58</td>
</tr>
<tr>
<td>XML Config File Panel</td>
<td>45</td>
</tr>
</tbody>
</table>