Copyright

Copyright © 2012-2014 Intergraph® Corporation. All Rights Reserved. Intergraph is part of Hexagon.

Including software, file formats, and audiovisual displays; may be used pursuant to applicable software license agreement; contains confidential and proprietary information of Intergraph and/or third parties which is protected by copyright law, trade secret law, and international treaty, and may not be provided or otherwise made available without proper authorization from Intergraph Corporation.

U.S. Government Restricted Rights Legend

Use, duplication, or disclosure by the government is subject to restrictions as set forth below. For civilian agencies: This was developed at private expense and is "restricted computer software" submitted with restricted rights in accordance with subparagraphs (a) through (d) of the Commercial Computer Software - Restricted Rights clause at 52.227-19 of the Federal Acquisition Regulations ("FAR") and its successors, and is unpublished and all rights are reserved under the copyright laws of the United States. For units of the Department of Defense ("DoD"): This is "commercial computer software" as defined at DFARS 252.227-7014 and the rights of the Government are as specified at DFARS 227.7202-3.

Unpublished - rights reserved under the copyright laws of the United States.

Intergraph Corporation
300 Intergraph Way
Huntsville, AL 35813

Documentation


Other Documentation

Other Documentation shall mean, whether in electronic or printed form and delivered with software or on Intergraph Smart Support, SharePoint, or box.net, any documentation related to work processes, workflows, and best practices that is provided by Intergraph as guidance for using a software product.

Terms of Use

a. Use of a software product and Documentation is subject to the End User License Agreement ("EULA") delivered with the software product unless the Licensee has a valid signed license for this software product with Intergraph Corporation. If the Licensee has a valid signed license for this software product with Intergraph Corporation, the valid signed license shall take precedence and govern the use of this software product and Documentation. Subject to the terms contained within the applicable license agreement, Intergraph Corporation gives Licensee permission to print a reasonable number of copies of the Documentation as defined in the applicable license agreement and delivered with the software product for Licensee's internal, non-commercial use. The Documentation may not be printed for resale or redistribution.

b. For use of Documentation or Other Documentation where end user does not receive a EULA or does not have a valid license agreement with Intergraph, Intergraph grants the Licensee a non-exclusive license to use the Documentation or Other Documentation for Licensee's internal non-commercial use. Intergraph Corporation gives Licensee permission to print a reasonable number of copies of Other Documentation for Licensee's internal, non-commercial. The Other Documentation may not be printed for resale or redistribution. This license contained in this subsection b) may be terminated at any time and for any reason by Intergraph Corporation by giving written notice to Licensee.

Disclaimer of Warranties

Except for any express warranties as may be stated in the EULA or separate license or separate terms and conditions, Intergraph Corporation disclaims any and all express or implied warranties including, but not limited to the implied warranties of merchantability and fitness for a particular purpose and nothing stated in, or implied by, this document or its contents shall be considered or deemed a modification or amendment of such disclaimer. Intergraph believes the information in this publication is accurate as of its publication date.

The information and the software discussed in this document are subject to change without notice and are subject to applicable technical product descriptions. Intergraph Corporation is not responsible for any error that may appear in this document.

The software, Documentation and Other Documentation discussed in this document are furnished under a license and may be used or copied only in accordance with the terms of this license. THE USER OF THE SOFTWARE IS EXPECTED TO MAKE THE FINAL EVALUATION AS TO THE USEFULNESS OF THE SOFTWARE IN HIS OWN ENVIRONMENT.

Intergraph is not responsible for the accuracy of delivered data including, but not limited to, catalog, reference and symbol data. Users should verify for themselves that the data is accurate and suitable for their project work.
Limitation of Damages

IN NO EVENT WILL INTERGRAPH CORPORATION BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL INCIDENTAL, SPECIAL, OR PUNITIVE DAMAGES, INCLUDING BUT NOT LIMITED TO, LOSS OF USE OR PRODUCTION, LOSS OF REVENUE OR PROFIT, LOSS OF DATA, OR CLAIMS OF THIRD PARTIES, EVEN IF INTERGRAPH CORPORATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

UNDER NO CIRCUMSTANCES SHALL INTERGRAPH CORPORATION’S LIABILITY EXCEED THE AMOUNT THAT INTERGRAPH CORPORATION HAS BEEN PAID BY LICENSEE UNDER THIS AGREEMENT AT THE TIME THE CLAIM IS MADE. EXCEPT WHERE PROHIBITED BY APPLICABLE LAW, NO CLAIM, REGARDLESS OF FORM, ARISING OUT OF OR IN CONNECTION WITH THE SUBJECT MATTER OF THIS DOCUMENT MAY BE BROUGHT BY LICENSEE MORE THAN TWO (2) YEARS AFTER THE EVENT GIVING RISE TO THE CAUSE OF ACTION HAS OCCURRED.

IF UNDER THE LAW RULED APPLICABLE ANY PART OF THIS SECTION IS INVALID, THEN INTERGRAPH LIMITS ITS LIABILITY TO THE MAXIMUM EXTENT ALLOWED BY SAID LAW.

Export Controls

Intergraph Corporation’s software products and any third-party Software Products obtained from Intergraph Corporation, its subsidiaries, or distributors (including any Documentation, Other Documentation or technical data related to these products) are subject to the export control laws and regulations of the United States. Diversion contrary to U.S. law is prohibited. These Software Products, and the direct product thereof, must not be exported or re-exported, directly or indirectly (including via remote access) under the following circumstances:

a. To Cuba, Iran, North Korea, Sudan, or Syria, or any national of these countries.
c. To any entity when Licensee knows, or has reason to know, the end use of the Software Product is related to the design, development, production, or use of missiles, chemical, biological, or nuclear weapons, or other un-safeguarded or sensitive nuclear uses.
d. To any entity when Licensee knows, or has reason to know, that an illegal reshipment will take place.
e. Any questions regarding export or re-export of these Software Products should be addressed to Intergraph Corporation’s Export Compliance Department, Huntsville, Alabama 35894, USA.

Trademarks

Intergraph, the Intergraph logo, Intergraph Smart, PDS, SmartPlant, SmartMarine, FrameWorks, I-Sketch, IntelliShip, ISOGEN, SmartSketch, SPOOLGEN, SupportManager, SupportModeler, Sapphire, and FreeView are trademarks or registered trademarks of Intergraph Corporation or its subsidiaries in the United States and other countries. Microsoft and Windows are registered trademarks of Microsoft Corporation. MicroStation is a registered trademark of Bentley Systems, Inc. Other brands and product names are trademarks of their respective owners.
Contents

Preface .......................................................................................................................................................... 21
  Documentation Comments ........................................................................................................................ 21

Overview .................................................................................................................................................. 23

Maintain Customer CIPs .......................................................................................................................... 25
  SQL*Plus ............................................................................................................................................. 25
  Development Tool ............................................................................................................................... 26
  A.60.37 Customer CIPs/Forms/Reports ............................................................................................. 26

Maintain CIP Relations ............................................................................................................................. 31

View CIP Relations ................................................................................................................................. 33

CIP Example ........................................................................................................................................... 35
  Prerequisites ....................................................................................................................................... 35
  Create the CIP Procedure .................................................................................................................. 35
  Use the Compiled CIP .......................................................................................................................... 37

Custom Packages ................................................................................................................................. 39
  M_PCK_ACCOUNT_CUSTOM .................................................................................................................. 39
    ACCOUNT_QUERY ............................................................................................................................ 39
    CALC_DISPO ................................................................................................................................. 40
    CALC_ORDER_ACCOUNTS ............................................................................................................... 40
    CALC_RELEASE .............................................................................................................................. 40
    CHECK_BUDGET_PO ..................................................................................................................... 41
    CHECK_BUDGET_REQ .................................................................................................................... 41
    DISTRIBUTE_OTHER_COSTS ......................................................................................................... 41
    FETCH_PROGRESS ......................................................................................................................... 42
    GET_BUDGET_DATA ......................................................................................................................... 42
  M_PCK_APPROVAL_CUSTOM ................................................................................................................ 43
    BUILD_APPROVAL_BODY ................................................................................................................ 43
    GET_APPROVAL_TEXT1 .................................................................................................................... 43
    GET_APPROVAL_TEXT2 .................................................................................................................... 45
    GET_APPROVAL_TEXT3 .................................................................................................................... 45
    GET_APPROVAL_TEXT4 .................................................................................................................... 45
  M_PCK_BC_CUSTOM ............................................................................................................................. 46
    BOM_DISPO_QTY ............................................................................................................................ 46
    GET_ER_QTY_TOTAL ........................................................................................................................ 46
    GET_ONLY_PO_QTY_TOTAL ............................................................................................................. 46
    GET_PO_OR_ER_QTY_TOTAL ............................................................................................................ 47
    SET_QUANTITY ............................................................................................................................... 47
    SET_SELECT .................................................................................................................................... 47
  M_PCK_BIR_CUSTOM ............................................................................................................................ 48
    GET_ACTUAL_SHIP_DATE .................................................................................................................. 48
    GET_BIR_CONVERTED_VALUE ......................................................................................................... 48
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>M_PCK_INVOICE_CUSTOM</td>
<td>83</td>
</tr>
<tr>
<td>GET_TABLE_DETAIL_INFO</td>
<td>68</td>
</tr>
<tr>
<td>GET_SUBJECT_BODY_ANSWER</td>
<td>70</td>
</tr>
<tr>
<td>POST_LOAD</td>
<td>78</td>
</tr>
<tr>
<td>POST_IMPORT</td>
<td>77</td>
</tr>
<tr>
<td>PDF_FILE_NAME</td>
<td>71</td>
</tr>
<tr>
<td>ZIP_FILE_NAME</td>
<td>73</td>
</tr>
<tr>
<td>REPORT_EXAMPLE</td>
<td>71</td>
</tr>
<tr>
<td>REPORT_EXAMPLE2</td>
<td>72</td>
</tr>
<tr>
<td>REPORT_EXAMPLE3</td>
<td>72</td>
</tr>
<tr>
<td>REPORT_EXAMPLE4</td>
<td>72</td>
</tr>
<tr>
<td>ZIPNAME_INIT</td>
<td>73</td>
</tr>
<tr>
<td>ZIPNAME_INIT</td>
<td>73</td>
</tr>
<tr>
<td>M_PCK_EXP_CUSTOM</td>
<td>73</td>
</tr>
<tr>
<td>POST_IE_ATTR</td>
<td>75</td>
</tr>
<tr>
<td>POST_RELN_APPROVAL</td>
<td>76</td>
</tr>
<tr>
<td>POST_RELN_REV_APPROVAL</td>
<td>76</td>
</tr>
<tr>
<td>POST_RELN_APPROVAL</td>
<td>77</td>
</tr>
<tr>
<td>PRE_RELN_APPROVAL</td>
<td>77</td>
</tr>
<tr>
<td>M_PCK_FAB_CUSTOM</td>
<td>77</td>
</tr>
<tr>
<td>POST_IMPORT</td>
<td>77</td>
</tr>
<tr>
<td>POST_LOAD</td>
<td>78</td>
</tr>
<tr>
<td>POST_LOAD</td>
<td>78</td>
</tr>
<tr>
<td>UPDATE_FT</td>
<td>79</td>
</tr>
<tr>
<td>M_PCK_FORECAST_CUSTOM</td>
<td>79</td>
</tr>
<tr>
<td>CHECK_BEFORE_RR</td>
<td>79</td>
</tr>
<tr>
<td>CHECK_RESERVE</td>
<td>80</td>
</tr>
<tr>
<td>COST</td>
<td>80</td>
</tr>
<tr>
<td>MANHOURS</td>
<td>80</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>81</td>
</tr>
<tr>
<td>M_PCK_IDENT_CUSTOM</td>
<td>81</td>
</tr>
<tr>
<td>POST_BATCH</td>
<td>81</td>
</tr>
<tr>
<td>PRE_BATCH</td>
<td>81</td>
</tr>
<tr>
<td>START_BATCH_BUILD</td>
<td>82</td>
</tr>
<tr>
<td>M_PCK_INTOOLS_CUSTOM</td>
<td>82</td>
</tr>
<tr>
<td>ADD_ATTRS</td>
<td>82</td>
</tr>
<tr>
<td>PRE_LOADING</td>
<td>82</td>
</tr>
<tr>
<td>M_PCK_INVOICE_CUSTOM</td>
<td>83</td>
</tr>
<tr>
<td>GET_IVC_NUMBER</td>
<td>83</td>
</tr>
<tr>
<td>Procedure</td>
<td>Page</td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>M_PCK_OMI_CUSTOM</td>
<td>104</td>
</tr>
<tr>
<td>M_PCK_MTI_CUSTOM</td>
<td>103</td>
</tr>
<tr>
<td>M_PCK_MS_CUSTOM</td>
<td>101</td>
</tr>
<tr>
<td>M_PCK_MENU_DICT_CUSTOM</td>
<td>100</td>
</tr>
<tr>
<td>M_PCK_MAIL_CUSTOM</td>
<td>99</td>
</tr>
<tr>
<td>GET_REMIT_ADD</td>
<td>83</td>
</tr>
<tr>
<td>SUBMIT_INVOICE</td>
<td>84</td>
</tr>
<tr>
<td>UNSUBMIT_INVOICE</td>
<td>84</td>
</tr>
<tr>
<td>UPD_INVOICE_STATUS</td>
<td>85</td>
</tr>
<tr>
<td>M_PCK_ISKETCH_CUSTOM</td>
<td>86</td>
</tr>
<tr>
<td>CONVERT_BORE_UNIT_TO_UNIT</td>
<td>86</td>
</tr>
<tr>
<td>CONVERT_DIM_UNIT_TO_UNIT</td>
<td>87</td>
</tr>
<tr>
<td>GENERAL_CIP</td>
<td>88</td>
</tr>
<tr>
<td>GET_CC_DESCRIPTION</td>
<td>89</td>
</tr>
<tr>
<td>GET_IDENT_BOLT_DIMS</td>
<td>89</td>
</tr>
<tr>
<td>GET_IDENT_DESCRIPTION</td>
<td>90</td>
</tr>
<tr>
<td>M_PCK_LOGISTIC_CUSTOM</td>
<td>90</td>
</tr>
<tr>
<td>GET_IS_POS</td>
<td>91</td>
</tr>
<tr>
<td>GET_ORDER_NUMBER</td>
<td>91</td>
</tr>
<tr>
<td>GET_SG_CODE</td>
<td>91</td>
</tr>
<tr>
<td>GET_SHIPPING_STATUS</td>
<td>92</td>
</tr>
<tr>
<td>P7061_SAMPLE</td>
<td>92</td>
</tr>
<tr>
<td>P7062_SAMPLE</td>
<td>93</td>
</tr>
<tr>
<td>M_PCK_MAIL_CUSTOM</td>
<td>93</td>
</tr>
<tr>
<td>CREATE_MSG_TEXT_INQ</td>
<td>93</td>
</tr>
<tr>
<td>CREATE_MSG_TEXT_SUP</td>
<td>94</td>
</tr>
<tr>
<td>CREATE_MSG_TEXT_ADD</td>
<td>94</td>
</tr>
<tr>
<td>CREATE_MSG_TEXT_PO</td>
<td>94</td>
</tr>
<tr>
<td>CREATE_MSG_TEXT_Q</td>
<td>95</td>
</tr>
<tr>
<td>CREATE_MSG_TEXT_REQ</td>
<td>95</td>
</tr>
<tr>
<td>CREATE_MSG_TEXT_QO</td>
<td>95</td>
</tr>
<tr>
<td>CREATE_MSG_TEXT_PNO</td>
<td>96</td>
</tr>
<tr>
<td>CREATE_MSG_TEXT_SOV</td>
<td>96</td>
</tr>
<tr>
<td>CREATE_MSG_TEXT_COR</td>
<td>96</td>
</tr>
<tr>
<td>CREATE_MSG_TEXT_BC</td>
<td>97</td>
</tr>
<tr>
<td>CREATE_MSG_TEXT_NCR</td>
<td>97</td>
</tr>
<tr>
<td>CREATE_MSG_TEXT_CLOI</td>
<td>97</td>
</tr>
<tr>
<td>CREATE_MSG_TEXT_INV</td>
<td>98</td>
</tr>
<tr>
<td>CREATE_MSG_TEXT_LR</td>
<td>98</td>
</tr>
<tr>
<td>CREATE_MSG_TEXT_CP</td>
<td>98</td>
</tr>
<tr>
<td>CREATE_MSG_TEXT_MIR</td>
<td>99</td>
</tr>
<tr>
<td>GET_REVERSE_BODY_TEXT</td>
<td>99</td>
</tr>
<tr>
<td>GET_REVERSE_SUBJECT_TEXT</td>
<td>99</td>
</tr>
<tr>
<td>SEND_NOTIFICATIONS</td>
<td>100</td>
</tr>
<tr>
<td>M_PCK_MENU_DICT_CUSTOM</td>
<td>100</td>
</tr>
<tr>
<td>POPULATE_AUTOCORRECT_TAB</td>
<td>100</td>
</tr>
<tr>
<td>POPULATE_MISSPELL_TAB</td>
<td>101</td>
</tr>
<tr>
<td>M_PCK_MS_CUSTOM</td>
<td>101</td>
</tr>
<tr>
<td>CALC_ACTUAL_DATE_MASTER</td>
<td>101</td>
</tr>
<tr>
<td>CALC_PLANNED_ISSUE_DATE</td>
<td>102</td>
</tr>
<tr>
<td>GET_MS_VAL</td>
<td>102</td>
</tr>
<tr>
<td>SAMPLE_FNC</td>
<td>102</td>
</tr>
<tr>
<td>SET_MS_PLANNING_DATE</td>
<td>103</td>
</tr>
<tr>
<td>M_PCK_MTI_CUSTOM</td>
<td>103</td>
</tr>
<tr>
<td>FILL_PO_UDF</td>
<td>103</td>
</tr>
<tr>
<td>FILL_REQ_UDF</td>
<td>104</td>
</tr>
<tr>
<td>M_PCK_OMI_CUSTOM</td>
<td>104</td>
</tr>
<tr>
<td>BOM_NODE_CREATED</td>
<td>104</td>
</tr>
<tr>
<td>GET_SHIPPING_STATUS</td>
<td>92</td>
</tr>
<tr>
<td>GET_SG_CODE</td>
<td>91</td>
</tr>
<tr>
<td>GET_ORDER_NUMBER</td>
<td>91</td>
</tr>
<tr>
<td>GET_SG_CODE</td>
<td>91</td>
</tr>
<tr>
<td>GET_SHIPPING_STATUS</td>
<td>92</td>
</tr>
<tr>
<td>P7061_SAMPLE</td>
<td>92</td>
</tr>
<tr>
<td>P7062_SAMPLE</td>
<td>93</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>BOM_NODE_FOUND</td>
<td>105</td>
</tr>
<tr>
<td>BOM_NODE_PRE_TSQ</td>
<td>105</td>
</tr>
<tr>
<td>BOM_POS_CREATED</td>
<td>105</td>
</tr>
<tr>
<td>CHECK_CONSISTENCY</td>
<td>106</td>
</tr>
<tr>
<td>OMI_EDIT_FLAT_PRE_IMPORTS</td>
<td>106</td>
</tr>
<tr>
<td>OMI_FIND_CC_BY_TD</td>
<td>106</td>
</tr>
<tr>
<td>OMI_FLC_LOAD</td>
<td>107</td>
</tr>
<tr>
<td>OMI_JOB_AFTER_COPY</td>
<td>107</td>
</tr>
<tr>
<td>OMI_JOB_END</td>
<td>107</td>
</tr>
<tr>
<td>OMI_JOB_START</td>
<td>108</td>
</tr>
<tr>
<td>OMI_LOAD</td>
<td>108</td>
</tr>
<tr>
<td>OMI_POST_CHANGES_AFTER_IMPORT</td>
<td>108</td>
</tr>
<tr>
<td>OMI_SET_ITEM_TYPE</td>
<td>109</td>
</tr>
<tr>
<td>OMI_SET_UNIT</td>
<td>109</td>
</tr>
<tr>
<td>M_PCK_OPI_CUSTOM</td>
<td>109</td>
</tr>
<tr>
<td>ASCII_DATE</td>
<td>110</td>
</tr>
<tr>
<td>DATA_DATE</td>
<td>110</td>
</tr>
<tr>
<td>POST_OPI_IMPORT</td>
<td>110</td>
</tr>
<tr>
<td>XML_DATE</td>
<td>111</td>
</tr>
<tr>
<td>M_PCK_OSCI_CUSTOM</td>
<td>111</td>
</tr>
<tr>
<td>XML_DATE</td>
<td>111</td>
</tr>
<tr>
<td>M_PCK_PDMSEX_DP_CUSTOM</td>
<td>112</td>
</tr>
<tr>
<td>PUBLISH</td>
<td>112</td>
</tr>
<tr>
<td>M_PCK_PDMS_CUSTOM</td>
<td>112</td>
</tr>
<tr>
<td>BFLE_WEIGHT</td>
<td>112</td>
</tr>
<tr>
<td>BOLT_HANDLING</td>
<td>112</td>
</tr>
<tr>
<td>CATE</td>
<td>113</td>
</tr>
<tr>
<td>CATE_STEXT</td>
<td>113</td>
</tr>
<tr>
<td>CATE_TEXT</td>
<td>114</td>
</tr>
<tr>
<td>CHECK_EXPORT_ALLOWED</td>
<td>114</td>
</tr>
<tr>
<td>CHECK_IDENTS</td>
<td>114</td>
</tr>
<tr>
<td>CLEAN_SORT_ALONG_LINE</td>
<td>115</td>
</tr>
<tr>
<td>COCO</td>
<td>115</td>
</tr>
<tr>
<td>DTSE</td>
<td>115</td>
</tr>
<tr>
<td>example_cip_pdms3002</td>
<td>116</td>
</tr>
<tr>
<td>FILE_BOLT_REF</td>
<td>116</td>
</tr>
<tr>
<td>FILL_COMP_ATTRS</td>
<td>116</td>
</tr>
<tr>
<td>FIND_IN_CCD</td>
<td>117</td>
</tr>
<tr>
<td>GENERAL_PDMS2001</td>
<td>117</td>
</tr>
<tr>
<td>GENERAL_SPEC_FILE_UPDATE</td>
<td>117</td>
</tr>
<tr>
<td>GET_ATTR_VALUE</td>
<td>118</td>
</tr>
<tr>
<td>GET_BTYPE</td>
<td>118</td>
</tr>
<tr>
<td>GET_CATA</td>
<td>118</td>
</tr>
<tr>
<td>GET_FLPB</td>
<td>119</td>
</tr>
<tr>
<td>GET_IDENT_CODE</td>
<td>119</td>
</tr>
<tr>
<td>GET_INSU_CATREF</td>
<td>119</td>
</tr>
<tr>
<td>GET_INSU_DETAIL</td>
<td>120</td>
</tr>
<tr>
<td>GET_INSU_DETAIL</td>
<td>120</td>
</tr>
<tr>
<td>GET_INSU_NAME</td>
<td>121</td>
</tr>
<tr>
<td>GET_INSU_PBOR0</td>
<td>121</td>
</tr>
<tr>
<td>GET_INSU_TEMP</td>
<td>121</td>
</tr>
<tr>
<td>GET_INSU_TYPE</td>
<td>122</td>
</tr>
<tr>
<td>Function</td>
<td>Page</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>GET_PDMNS_INTERFACE_ID</td>
<td>122</td>
</tr>
<tr>
<td>GET_UNIT</td>
<td>122</td>
</tr>
<tr>
<td>GPD_REF</td>
<td>123</td>
</tr>
<tr>
<td>GTYPE</td>
<td>123</td>
</tr>
<tr>
<td>IDENT_ATTRS</td>
<td>123</td>
</tr>
<tr>
<td>PREP_OUT_SET_DELETE_DESC</td>
<td>124</td>
</tr>
<tr>
<td>PREP_OUTPUT</td>
<td>124</td>
</tr>
<tr>
<td>PREP_OUTPUT_DEL</td>
<td>124</td>
</tr>
<tr>
<td>PREP_OUTPUT_HEADING</td>
<td>125</td>
</tr>
<tr>
<td>PREP_OUTPUT_INS</td>
<td>125</td>
</tr>
<tr>
<td>PREP_OUTPUT_MOD</td>
<td>125</td>
</tr>
<tr>
<td>S2008</td>
<td>126</td>
</tr>
<tr>
<td>SCOM</td>
<td>126</td>
</tr>
<tr>
<td>SECT</td>
<td>126</td>
</tr>
<tr>
<td>SET_INSU_FIELDS</td>
<td>127</td>
</tr>
<tr>
<td>SET_S2008</td>
<td>127</td>
</tr>
<tr>
<td>SET_SIZE_CONVERSION</td>
<td>127</td>
</tr>
<tr>
<td>SKEY</td>
<td>128</td>
</tr>
<tr>
<td>SORT_ALONG_LINE</td>
<td>128</td>
</tr>
<tr>
<td>SPEC_BOLT_REF</td>
<td>128</td>
</tr>
<tr>
<td>TEXT</td>
<td>129</td>
</tr>
<tr>
<td>M_PCK_PDSEXPA_CUSTOM</td>
<td>129</td>
</tr>
<tr>
<td>PUBLISH</td>
<td>129</td>
</tr>
<tr>
<td>M_PCK_PDS_BOM_CUSTOM</td>
<td>130</td>
</tr>
<tr>
<td>CREATE_OR_SUM_SPECIAL_ITEMS</td>
<td>130</td>
</tr>
<tr>
<td>GET_LN_CODE</td>
<td>130</td>
</tr>
<tr>
<td>INSERT_NODE_NLS</td>
<td>130</td>
</tr>
<tr>
<td>M_PCK_PDS_CUSTOM</td>
<td>131</td>
</tr>
<tr>
<td>AFTER_PDL_GENERATION</td>
<td>131</td>
</tr>
<tr>
<td>BOLT_LOGIC</td>
<td>131</td>
</tr>
<tr>
<td>COMMODITY_DESCRIPTION</td>
<td>131</td>
</tr>
<tr>
<td>GASKET_LOGIC</td>
<td>132</td>
</tr>
<tr>
<td>GET_COMMODITY_CODE</td>
<td>132</td>
</tr>
<tr>
<td>GET_ENDTYPE</td>
<td>133</td>
</tr>
<tr>
<td>GET_FL_WITH_IO_LINE_IND</td>
<td>133</td>
</tr>
<tr>
<td>GET_PDLH_CC</td>
<td>133</td>
</tr>
<tr>
<td>GET_PDLH_ENDPRP</td>
<td>134</td>
</tr>
<tr>
<td>GET_PDLH_GEOSTD</td>
<td>134</td>
</tr>
<tr>
<td>GET_PDLH_MATERIAL</td>
<td>134</td>
</tr>
<tr>
<td>GET_PDLH_MODEL</td>
<td>135</td>
</tr>
<tr>
<td>GET_PDLH_MODIF</td>
<td>135</td>
</tr>
<tr>
<td>GET_PDLH_RATING</td>
<td>135</td>
</tr>
<tr>
<td>GET_PDLH_SCHEDULE</td>
<td>136</td>
</tr>
<tr>
<td>GET_PDLH_TS</td>
<td>136</td>
</tr>
<tr>
<td>GET_PDLH_WR</td>
<td>136</td>
</tr>
<tr>
<td>GET_SDD_LINE</td>
<td>137</td>
</tr>
<tr>
<td>IDENT_DESCRIPTION</td>
<td>137</td>
</tr>
<tr>
<td>M_PCK_PID_SPEC.TXT_CUSTOM</td>
<td>137</td>
</tr>
<tr>
<td>GET_COMMODITYOPTION_LONGTXT</td>
<td>137</td>
</tr>
<tr>
<td>GET_COMMODITYOPTION_SHORTTXT</td>
<td>138</td>
</tr>
<tr>
<td>GET_MAXTEMP</td>
<td>138</td>
</tr>
<tr>
<td>GET_MINMAXTEMP</td>
<td>139</td>
</tr>
<tr>
<td>GET_SCHEDULE_CDLST</td>
<td>140</td>
</tr>
</tbody>
</table>
M_PCK_QUERY_CUSTOM ............................................................................................................... 153
M_PCK_PLL_VERIFY_CUSTOM....................................................................................................... 143
M_PCK_PLATE_CUSTOM ................................................................................................................. 142
GET_FIFTH_AWARDING_VALUE .............................................................................................. 157
GET_EIGHTH_PROMPT ............................................................................................................. 156
GET_COMMODITY_CODE_PROMPT ........................................................................................ 156
GET_CE_MISSING_PROMPT .................................................................................................... 155
GET_ALT_PROMPT .................................................................................................................... 155
AFTER_CHECK_INQ_PRECON ................................................................................................. 154
CHECK_COND ............................................................................................................................ 154
M_PCK_QUERY_CUSTOMER ........................................................................................................ 154
EXECUTE_CIP ............................................................................................................................. 148
EXEC_GENERAL_CIP................................................................................................................. 149
GEN_INQ_NUMBER................................................................. 149
GEN_ORDER_NUMBER................................................................. 150
GET_PO_STATUS................................................................. 150
GET_SUPPL_DELV_STATUS................................................................. 150
GET_SUPPL_STATUS................................................................. 151
IMPORT_PB_ITEMS_CIP................................................................. 151
POST_APPLY_PRICES................................................................. 151
POST_APPROVAL................................................................. 152
POST_POH_CREATION................................................................. 152
POST_REVERSE_APPROVAL................................................................. 152
POST_SET_ISSUE_DATE................................................................. 153
RENUMBER_POS................................................................. 153
M_PCK_PO_CUSTOM.................................................................................................................. 145
AGREEMENT_APPROVAL................................................................. 145
BEFORE_POH_CREATION................................................................. 146
CHECK_PO_NUMBER................................................................. 146
CHECK_PRINT_ORDER................................................................. 146
CHECK_REV_APP_ALLOWED................................................................. 147
CHECK_RFA................................................................. 147
CHECK_SET_ISSUE_DATE................................................................. 147
DEFAULT_ACCOUNT................................................................. 148
DELETE_ORDER................................................................. 148
EXECUTE_CIP................................................................. 148
GET_SUPPL_DELV_STATUS................................................................. 150
GET_PO_STATUS................................................................. 150
GET_SUPPL_STATUS................................................................. 151
IMPORT_PB_ITEMS_CIP................................................................. 151
POST_APPLY_PRICES................................................................. 151
POST_APPROVAL................................................................. 152
POST_POH_CREATION................................................................. 152
POST_REVERSE_APPROVAL................................................................. 152
POST_SET_ISSUE_DATE................................................................. 153
RENUMBER_POS................................................................. 153
M_PCK_QUERY_CUSTOM........................................................................................................ 154
CHECK_COND ............................................................................................................................ 154
M_PCK_QUOTE_CUSTOM................................................................. 154
AFTER_CHECK_INQ_PRECON................................................................. 154
BEFORE_APPROVAL................................................................. 154
CHECK_BEFORE_INQ_APPR................................................................. 155
GET_ALT_PROMPT................................................................. 155
GET_CE_MISSING_PROMPT................................................................. 155
GET_COMMODITY_CODE_PROMPT................................................................. 156
GET_EIGHTH_AWARDING_VALUE................................................................. 156
GET_EIGHTH_PROMPT................................................................. 156
GET_EIGHTH_VALUE................................................................. 157
GET_FIFTH_AWARDING_SUM................................................................. 157
GET_FIFTH_AWARDING_VALUE................................................................. 157
GET_FIFTH_FORMAT................................................................. 158
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET_FIFTH_PROMPT</td>
<td>158</td>
</tr>
<tr>
<td>GET_FIFTH_SUM_LINE</td>
<td>158</td>
</tr>
<tr>
<td>GET_FIFTH_SUM_LINE_FORMAT</td>
<td>159</td>
</tr>
<tr>
<td>GET_FIFTH_SUM_LINE_PROMPT</td>
<td>159</td>
</tr>
<tr>
<td>GET_FIFTH_VALUE</td>
<td>159</td>
</tr>
<tr>
<td>GET_FIRST_AWARDING_SUM</td>
<td>160</td>
</tr>
<tr>
<td>GET_FIRST_AWARDING_VALUE</td>
<td>160</td>
</tr>
<tr>
<td>GET_FIRST_FORMAT</td>
<td>160</td>
</tr>
<tr>
<td>GET_FIRST_PROMPT</td>
<td>161</td>
</tr>
<tr>
<td>GET_FIRST_SUM_LINE</td>
<td>161</td>
</tr>
<tr>
<td>GET_FIRST_SUM_LINE_FORMAT</td>
<td>161</td>
</tr>
<tr>
<td>GET_FIRST_SUM_LINE_PROMPT</td>
<td>162</td>
</tr>
<tr>
<td>GET_FIRST_TOTAL_FORMAT</td>
<td>162</td>
</tr>
<tr>
<td>GET_FIRST_TOTAL_PROMPT</td>
<td>162</td>
</tr>
<tr>
<td>GET_FIRST_TOTAL_VALUE</td>
<td>163</td>
</tr>
<tr>
<td>GET_FIRST_VALUE</td>
<td>163</td>
</tr>
<tr>
<td>GET_FIRST_VB_HEADER</td>
<td>163</td>
</tr>
<tr>
<td>GET_FOURTH_AWARDING_SUM</td>
<td>164</td>
</tr>
<tr>
<td>GET_FOURTH_AWARDING_VALUE</td>
<td>164</td>
</tr>
<tr>
<td>GET_FOURTH_FORMAT</td>
<td>164</td>
</tr>
<tr>
<td>GET_FOURTH_PROMPT</td>
<td>165</td>
</tr>
<tr>
<td>GET_FOURTH_SUM_LINE</td>
<td>165</td>
</tr>
<tr>
<td>GET_FOURTH_SUM_LINE_FORMAT</td>
<td>165</td>
</tr>
<tr>
<td>GET_FOURTH_SUM_LINE_PROMPT</td>
<td>166</td>
</tr>
<tr>
<td>GET_FOURTH_VALUE</td>
<td>166</td>
</tr>
<tr>
<td>GET_GROUP_BY_1_PROMPT</td>
<td>166</td>
</tr>
<tr>
<td>GET_GROUP_BY_2_PROMPT</td>
<td>167</td>
</tr>
<tr>
<td>GET_GROUP_BY_3_PROMPT</td>
<td>167</td>
</tr>
<tr>
<td>GET_GROUP_BY_4_PROMPT</td>
<td>167</td>
</tr>
<tr>
<td>GET_GROUP_BY_5_PROMPT</td>
<td>168</td>
</tr>
<tr>
<td>GET_IDENT_PROMPT</td>
<td>168</td>
</tr>
<tr>
<td>GET_IDENT_SHORT_PROMPT</td>
<td>168</td>
</tr>
<tr>
<td>GET_LINE_MARK</td>
<td>169</td>
</tr>
<tr>
<td>GET_NINTH_AWARDING_VALUE</td>
<td>169</td>
</tr>
<tr>
<td>GET_NINTH_PROMPT</td>
<td>169</td>
</tr>
<tr>
<td>GET_NINTH_VALUE</td>
<td>170</td>
</tr>
<tr>
<td>GET_NO_BID_PROMPT</td>
<td>170</td>
</tr>
<tr>
<td>GET_ORIGIN_PROMPT</td>
<td>170</td>
</tr>
<tr>
<td>GET_POS_PROMPT</td>
<td>171</td>
</tr>
<tr>
<td>GET_PROC_SEQ_PROMPT</td>
<td>171</td>
</tr>
<tr>
<td>GET_R_CODE_PROMPT</td>
<td>171</td>
</tr>
<tr>
<td>GET_R_SUPP_PROMPT</td>
<td>171</td>
</tr>
<tr>
<td>GET_SECOND_AWARDING_SUM</td>
<td>172</td>
</tr>
<tr>
<td>GET_SECOND_AWARDING_VALUE</td>
<td>172</td>
</tr>
<tr>
<td>GET_SECOND_FORMAT</td>
<td>173</td>
</tr>
<tr>
<td>GET_SECOND_PROMPT</td>
<td>173</td>
</tr>
<tr>
<td>GET_SECOND_SUM_LINE</td>
<td>173</td>
</tr>
<tr>
<td>GET_SECOND_SUM_LINE_FORMAT</td>
<td>174</td>
</tr>
<tr>
<td>GET_SECOND_SUM_LINE_PROMPT</td>
<td>174</td>
</tr>
<tr>
<td>GET_SECOND_TOTAL_FORMAT</td>
<td>174</td>
</tr>
<tr>
<td>GET_SECOND_TOTAL_PROMPT</td>
<td>174</td>
</tr>
<tr>
<td>GET_SECOND_TOTAL_VALUE</td>
<td>175</td>
</tr>
<tr>
<td>GET_SECOND_TOTAL_VALUE</td>
<td>175</td>
</tr>
<tr>
<td>GET_SECOND_VALUE</td>
<td>175</td>
</tr>
<tr>
<td>Procedure</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>M_PCK_REPORT_CUSTOM</td>
<td>188</td>
</tr>
<tr>
<td>SET_PRINTOUT</td>
<td>188</td>
</tr>
<tr>
<td>M_PCK_REQ_BLR_CUSTOM</td>
<td>188</td>
</tr>
<tr>
<td>ALLOW_REQ_DELETE</td>
<td>188</td>
</tr>
<tr>
<td>APPROVE_REQ</td>
<td>188</td>
</tr>
<tr>
<td>CALC_BUDGET</td>
<td>189</td>
</tr>
<tr>
<td>CHECK_FORCE_RTP</td>
<td>190</td>
</tr>
<tr>
<td>CREATE_PO_REQ_CIP</td>
<td>190</td>
</tr>
<tr>
<td>DESIGNATION</td>
<td>190</td>
</tr>
<tr>
<td>GENERAL_CIP</td>
<td>191</td>
</tr>
<tr>
<td>GET_IDENT_INQUIRY_QTY</td>
<td>191</td>
</tr>
<tr>
<td>GET_INQUIRY_QTY</td>
<td>192</td>
</tr>
<tr>
<td>IMPORT_PB_ITEMS_CIP</td>
<td>192</td>
</tr>
<tr>
<td>INSERT_M_REQS</td>
<td>192</td>
</tr>
<tr>
<td>NEW_SUPPLEMENT_CIP</td>
<td>193</td>
</tr>
<tr>
<td>RENUMBER_POS_SUB</td>
<td>193</td>
</tr>
<tr>
<td>SET_FORCE_RTP</td>
<td>193</td>
</tr>
<tr>
<td>Procedure Name</td>
<td>Page</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>SET_PROC_ORDER_SEQ</td>
<td>194</td>
</tr>
<tr>
<td>SET_REV_ATTR</td>
<td>194</td>
</tr>
<tr>
<td>STD_JOB_ID</td>
<td>195</td>
</tr>
<tr>
<td>STD_JOB_INIT</td>
<td>195</td>
</tr>
<tr>
<td>STD_PROC_CHANNEL</td>
<td>195</td>
</tr>
<tr>
<td>STD_REVERSE_REL</td>
<td>196</td>
</tr>
<tr>
<td>STD_ROS_DATE</td>
<td>196</td>
</tr>
<tr>
<td>SUM_INQUIRY</td>
<td>196</td>
</tr>
<tr>
<td>TECH_EVAL</td>
<td>197</td>
</tr>
<tr>
<td>UPDATE_INQ_REQ</td>
<td>197</td>
</tr>
<tr>
<td>M_PCK_REQ_CUSTOM</td>
<td>198</td>
</tr>
<tr>
<td>ALLOW_REQ_DELETION</td>
<td>198</td>
</tr>
<tr>
<td>APPROVE_RELEASE_REQ</td>
<td>198</td>
</tr>
<tr>
<td>APPROVE_REQ</td>
<td>199</td>
</tr>
<tr>
<td>APPROVE_REQ_BY</td>
<td>199</td>
</tr>
<tr>
<td>ATTACH_BOM_DOC_REQ_LINES_NODES</td>
<td>200</td>
</tr>
<tr>
<td>CALC_BUDGET</td>
<td>200</td>
</tr>
<tr>
<td>CHECK_FORCE_RTP</td>
<td>200</td>
</tr>
<tr>
<td>CHECK_RFA</td>
<td>201</td>
</tr>
<tr>
<td>CREATE_PO_REQ_CIP</td>
<td>201</td>
</tr>
<tr>
<td>DESIGNATION</td>
<td>202</td>
</tr>
<tr>
<td>GENERAL_CIP</td>
<td>202</td>
</tr>
<tr>
<td>GET_IDENT_INQUIRY_QTY</td>
<td>202</td>
</tr>
<tr>
<td>GET_INQUIRY_QTY</td>
<td>203</td>
</tr>
<tr>
<td>GET_ORDER</td>
<td>203</td>
</tr>
<tr>
<td>IMPORT_PB_ITEMS_CIP</td>
<td>203</td>
</tr>
<tr>
<td>INSERT_M_REQS</td>
<td>204</td>
</tr>
<tr>
<td>NEW_SUPPLEMENT_CIP</td>
<td>204</td>
</tr>
<tr>
<td>ORDER_REQ</td>
<td>204</td>
</tr>
<tr>
<td>QTY_UNIT</td>
<td>205</td>
</tr>
<tr>
<td>RENUMBER_POS_SUB</td>
<td>205</td>
</tr>
<tr>
<td>SET_FORCE_RTP</td>
<td>205</td>
</tr>
<tr>
<td>SET_PROC_ORDER_SEQ</td>
<td>206</td>
</tr>
<tr>
<td>SET_REV_ATTR</td>
<td>206</td>
</tr>
<tr>
<td>STD_DISPO_QTY</td>
<td>207</td>
</tr>
<tr>
<td>STD_GLOBAL_DISPO_QTY</td>
<td>207</td>
</tr>
<tr>
<td>STD_JOB_ID</td>
<td>207</td>
</tr>
<tr>
<td>STD_JOB_INIT</td>
<td>208</td>
</tr>
<tr>
<td>STD_PROC_CHANNEL</td>
<td>208</td>
</tr>
<tr>
<td>STD_REVERSE_REL</td>
<td>208</td>
</tr>
<tr>
<td>STD_ROS_DATE</td>
<td>209</td>
</tr>
<tr>
<td>SUM_INQUIRY</td>
<td>209</td>
</tr>
<tr>
<td>TECH_EVAL</td>
<td>209</td>
</tr>
<tr>
<td>UPDATE_INQ_REQ</td>
<td>210</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>210</td>
</tr>
<tr>
<td>WEIGHT_UNIT</td>
<td>211</td>
</tr>
<tr>
<td>M_PCK_S50R10_CUSTOM</td>
<td>211</td>
</tr>
<tr>
<td>GET_CORRAL</td>
<td>211</td>
</tr>
<tr>
<td>GET_FACING</td>
<td>211</td>
</tr>
<tr>
<td>GET_IDENT_SHORT</td>
<td>212</td>
</tr>
<tr>
<td>GET_MATL</td>
<td>212</td>
</tr>
<tr>
<td>GET_MAX_PRESS</td>
<td>212</td>
</tr>
<tr>
<td>GET_MAX_TEMP</td>
<td>213</td>
</tr>
</tbody>
</table>
Contents

GET_PLCNT_LOC ................................................................. 213
GET_PLOJ_NR ................................................................. 213
GET_RATING ................................................................. 214
GET_SERVICE ............................................................... 214
M_PCK_S50R11_CUSTOM .................................................. 214
GET_BRANCH_TABLE ......................................................... 214
GET_COMMODITY ........................................................ 215
GET_COMMODITY_DESC ................................................ 215
GET_CORRAL ............................................................... 215
GET_CTRL_SPC ............................................................ 216
GET_DSNSTD ............................................................... 216
GET_DETAIL_CONSTRUC ............................................... 216
GET_DETAIL_FINISH ...................................................... 217
GET_DETAIL_MATERIAL .................................................. 217
GET_DETAIL_RATING ..................................................... 217
GET_DETAIL_SCH .......................................................... 218
GET_END1 ................................................................. 218
GET_END2 ................................................................. 218
GET_FACING ............................................................... 219
GET_HEAT_SPEC ........................................................ 219
GET_MATL ................................................................. 219
GET_MAX_PRESS ........................................................... 220
GET_MAX_TEMP ........................................................... 220
GET_MIN_PRESS ........................................................... 220
GET_MIN_TEMP ........................................................... 221
GET_PLANT_LOC .......................................................... 221
GET_PLOJ_NR ............................................................... 221
GET_RATING ............................................................... 222
GET_SCHEDULE .......................................................... 222
GET_SERVICE ............................................................. 222
GET_SIZES ................................................................. 223
GET_THK ................................................................. 223
GET_WELDING_SPEC ..................................................... 223
M_PCK_SAP_CUSTOM ...................................................... 224
BUILD_SAP_ART_NR .................................................... 224
CHECK_TRANSFER ........................................................ 224
FETCH_ECLASS .......................................................... 224
FETCH_MATNR ............................................................ 225
FETCH_PLANT ............................................................. 225
GET_CHANGED_IDENTS .................................................. 225
GET_DESC ................................................................. 226
GET_GEWICHT ............................................................ 226
GET_IDENT_FROM_RLI .................................................. 226
GET_INTERFACE_ID ....................................................... 227
GET_LONG ............................................................... 227
GET_LONGDESC ........................................................ 227
GET_MEINS_NENNER ................................................... 228
GET_MEINS_ZAEHLER .................................................. 228
GET_REQ_DESCRIPTION ............................................... 229
GET_REQ_NAME .......................................................... 229
GET_RLI_IDENTDESC ..................................................... 229
GET_RLI_ROSDATE ....................................................... 230

SmartPlant Materials Customer-Installed Procedures (CIPs)
<table>
<thead>
<tr>
<th>Procedure Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET_RLI_UNIT</td>
<td>230</td>
</tr>
<tr>
<td>GET_RLI_WEIGHT_UNIT</td>
<td>230</td>
</tr>
<tr>
<td>GET_SHORTDESC</td>
<td>231</td>
</tr>
<tr>
<td>GET_SYSDATE</td>
<td>231</td>
</tr>
<tr>
<td>GET_TAG_TEXT</td>
<td>231</td>
</tr>
<tr>
<td>IDENT_CHANGE</td>
<td>232</td>
</tr>
<tr>
<td>M_PCK_SECURITY_CUSTOM</td>
<td>232</td>
</tr>
<tr>
<td>CHECK_COMPLEX_PASSWORD</td>
<td>232</td>
</tr>
<tr>
<td>GET_DISABLED_FK</td>
<td>232</td>
</tr>
<tr>
<td>GET_LAST_ANALYZED</td>
<td>233</td>
</tr>
<tr>
<td>M_PCK_SE_CUSTOM</td>
<td>233</td>
</tr>
<tr>
<td>FIELDSPIPE_SPEC</td>
<td>233</td>
</tr>
<tr>
<td>MAIN_MATERIAL</td>
<td>234</td>
</tr>
<tr>
<td>PROMPTPIPE_SPEC</td>
<td>234</td>
</tr>
<tr>
<td>TRAFFIC</td>
<td>235</td>
</tr>
<tr>
<td>M_PCK_SITE_CUSTOM</td>
<td>235</td>
</tr>
<tr>
<td>BEFORE_POST_MIR</td>
<td>235</td>
</tr>
<tr>
<td>CALCSELL PRICES</td>
<td>235</td>
</tr>
<tr>
<td>CHECKBEFORE_POST_MRR</td>
<td>236</td>
</tr>
<tr>
<td>CHECKBEFORE_POST_MTR</td>
<td>236</td>
</tr>
<tr>
<td>CHECKBEFORE_UNPOST</td>
<td>236</td>
</tr>
<tr>
<td>CHECKFIELDS</td>
<td>237</td>
</tr>
<tr>
<td>CHECKHEATNUMBERISSUE_QTY</td>
<td>237</td>
</tr>
<tr>
<td>CHECKHEATNUMBER_RECEIVED</td>
<td>237</td>
</tr>
<tr>
<td>CHECKMRR_RECV_QTY</td>
<td>238</td>
</tr>
<tr>
<td>CHECKOSDS</td>
<td>238</td>
</tr>
<tr>
<td>CHECK_VENDOR_CODE</td>
<td>238</td>
</tr>
<tr>
<td>ERPTRANSFER</td>
<td>239</td>
</tr>
<tr>
<td>ON_SITE_INS_STATUSTCHANGE</td>
<td>239</td>
</tr>
<tr>
<td>POSTDELETE_MRR</td>
<td>239</td>
</tr>
<tr>
<td>POST_MIR_CREATION</td>
<td>239</td>
</tr>
<tr>
<td>SITEINSPECTION</td>
<td>240</td>
</tr>
<tr>
<td>M_PCK_SM_CUSTOM</td>
<td>240</td>
</tr>
<tr>
<td>POSTAPPROVE_PERIOD_PROGRESS</td>
<td>241</td>
</tr>
<tr>
<td>POSTDELETEPERIOD_PROGRESS</td>
<td>241</td>
</tr>
<tr>
<td>POSTSUBMITPERIOD_PROGRESS</td>
<td>241</td>
</tr>
<tr>
<td>POSTUNAPPROVEPERIOD_PROGRESS</td>
<td>242</td>
</tr>
<tr>
<td>M_PCK_SP3DEXP_CUSTOM</td>
<td>242</td>
</tr>
<tr>
<td>PUBLISH</td>
<td>242</td>
</tr>
<tr>
<td>M_PCK_SP3D_CUSTOM</td>
<td>243</td>
</tr>
<tr>
<td>ADDCPN_STRING</td>
<td>243</td>
</tr>
<tr>
<td>ADDZERO</td>
<td>243</td>
</tr>
<tr>
<td>ANALYZELINE</td>
<td>243</td>
</tr>
<tr>
<td>BUILDCATALOGPARTNUMBER</td>
<td>244</td>
</tr>
<tr>
<td>CGRATTR_VALUE</td>
<td>244</td>
</tr>
<tr>
<td>CONVERTPHYDIM</td>
<td>244</td>
</tr>
<tr>
<td>CONVERTSIZE</td>
<td>245</td>
</tr>
<tr>
<td>CONVERTSIZEFROMASM</td>
<td>245</td>
</tr>
<tr>
<td>DELETEDUPVALVE_TAGIDENTS</td>
<td>245</td>
</tr>
<tr>
<td>FETCHCPNCUR</td>
<td>246</td>
</tr>
<tr>
<td>FILL_GENERICBOLTED</td>
<td>246</td>
</tr>
<tr>
<td>FILL_GENERICFEMALE</td>
<td>246</td>
</tr>
<tr>
<td>FILLPLAIN_GENERIC</td>
<td>247</td>
</tr>
</tbody>
</table>
GET_TD_DENSITY ...................................................................................................................... 263
GET_TABLE_DETAIL .................................................................................................................. 263
GET_STUBEND_CODE............................................................................................................... 262
GET_STUB_PATTERN ................................................................................................................ 262
GET_STUB_LENGTH .................................................................................................................. 262
GET_SPEC_HEADER_ID ............................................................................................................ 261
GET_SP3D_UOM ........................................................................................................................ 261
GET_SP3D_ATTR_VALUE ......................................................................................................... 261
GET_SI_SIZE_RANGE2 .............................................................................................................. 260
GET_SI_SIZE_RANGE ................................................................................................................ 260
GET_SI_MULTISIZEOPTION ...................................................................................................... 259
GET_SHT_NAME ......................................................................................................................... 259
GET_PORT_3 .............................................................................................................................. 258
GET_PORT .................................................................................................................................. 258
GET_NPS_SCH ........................................................................................................................... 258
GET_NPS_CONVERTED ............................................................................................................ 257
GET_MATERIAL_SHORT_DESC ............................................................................................... 257
GET_MATERIAL_LABEL_DESC ................................................................................................. 256
GET_MATERIAL_DESCRIPTION ............................................................................................... 256
GET_MASTER_CDLST_NUM ..................................................................................................... 256
GET_MAPPED_PARTCLASS ..................................................................................................... 255
GET_LOG_HEADER_PARM ....................................................................................................... 255
GET_IDENT_VALUE_PM ............................................................................................................ 255
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_SIZE_SCH .............................................................................................................. 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................................................................................................................... 254
GET_IDENT_VALUE .................
<p>| GET_TRANS_CC_GASKET                          | 265 |
| GET_TRANS_CC_GNW                            | 265 |
| GET_TRANS_CC_NUT                             | 265 |
| GET_TRANS_CC_WASHER                          | 266 |
| GET_UNIT_GSKTHK                              | 266 |
| GET_VALVE_CAT_NUM                            | 266 |
| GET_VC_ENDPREP_VALUE                        | 267 |
| GET_VC_ENDSTD_VALUE                         | 267 |
| GET_VC_NPD_VALUE                             | 267 |
| GET_VC_PRESSRAT_VALUE                       | 268 |
| GET_VC_SCH_VALUE                             | 268 |
| GET_VC_UNIT_VALUE                            | 268 |
| GET_WBK_TYPE                                | 269 |
| GET_XLS_ROW_NUMBER                          | 269 |
| IDENT_ON_SHEET                              | 269 |
| INIT_PARM_LIST                               | 270 |
| IS_GASKETCC                                  | 270 |
| IS_NUTCC                                     | 270 |
| IS_WASHERCC                                  | 271 |
| IS_PART_1SIZE_ONLY                          | 271 |
| MANIPULATE_MV_ITEM_IDENT_TEMP               | 271 |
| MULTISIZE_ACTION_FOR_PCF                    | 272 |
| OPEN_CPN_CUR                                | 272 |
| POST_PROCESS_FLTR_GEOM                      | 272 |
| POST_PROCESS_SHT_DATA                       | 273 |
| PRE_CHECK_FOR_SPEC_RULE_SHEETS              | 273 |
| RESET_SHT_CATEGORY                           | 273 |
| SAFE_TO_NUMBER                              | 274 |
| SET_ATTR_FOR_S3D_VERSION                    | 274 |
| SET_CATALOG_PART_NUMBER                     | 274 |
| SET_SHT_CATEGORY                             | 275 |
| SHT_CUST_METHOD                             | 275 |
| UPD_SI_SIZES_AND_SCHEDULES                  | 275 |
| M_PCK_SPEC_CUSTOM                            | 276 |
| CHECK_ISSUE_SPEC                             | 276 |
| CHK_SPEC_RATING_GEOM                        | 276 |
| EXECUTE_CUSTOMPROCEDURE                     | 277 |
| GENERAL_CIP                                  | 277 |
| GENERATE_SPEC_ITEMS                         | 278 |
| RELEASE_SPEC                                 | 278 |
| M_PCK_SPEC_REV_CUSTOM                       | 278 |
| BUILD_RELEASE_LIST                           | 278 |
| CHECK_FOR_UNRELEASE                          | 279 |
| CHECK_TEXT                                   | 279 |
| DELETE_SPEC_IDENTS                           | 279 |
| INSERT_SPEC_IDENTS                           | 280 |
| NEW_REV_GEOM_NAME                            | 280 |
| M_PCK_SPISO_CUSTOM                           | 280 |
| ComponentTypeForBendPipe                    | 280 |
| CONVERT_BORE_UNIT_TO_UNIT                   | 281 |
| CONVERT_DIM_UNIT_TO_UNIT                    | 282 |
| EXPORT_FILE_NAME                             | 282 |
| FORMAT_BOLT_CC                               | 283 |</p>
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Bend</td>
<td>283</td>
</tr>
<tr>
<td>General Branch</td>
<td>284</td>
</tr>
<tr>
<td>General CIP</td>
<td>284</td>
</tr>
<tr>
<td>General Default Choice</td>
<td>285</td>
</tr>
<tr>
<td>General Get Bolt</td>
<td>286</td>
</tr>
<tr>
<td>Get Bore Value Conv</td>
<td>286</td>
</tr>
<tr>
<td>Get CC Description</td>
<td>287</td>
</tr>
<tr>
<td>Get Dim Value Conv</td>
<td>287</td>
</tr>
<tr>
<td>Get Ident Bolt Dims</td>
<td>287</td>
</tr>
<tr>
<td>Get Ident Description</td>
<td>288</td>
</tr>
<tr>
<td>GetIdentWeight</td>
<td>288</td>
</tr>
<tr>
<td>SPEC_IDENTIFIER</td>
<td>289</td>
</tr>
<tr>
<td>M_PCK_STD_CUSTOM</td>
<td>289</td>
</tr>
<tr>
<td>CC_DESC</td>
<td>289</td>
</tr>
<tr>
<td>Check Branch</td>
<td>289</td>
</tr>
<tr>
<td>Check INV Ids</td>
<td>290</td>
</tr>
<tr>
<td>Check SHG</td>
<td>290</td>
</tr>
<tr>
<td>CMS IDENT CODE</td>
<td>290</td>
</tr>
<tr>
<td>CMS IDENT PROP VAL POST INSUPD</td>
<td>291</td>
</tr>
<tr>
<td>CODE_BLD_CIP</td>
<td>291</td>
</tr>
<tr>
<td>Commodity Code Check</td>
<td>291</td>
</tr>
<tr>
<td>Create Supplier</td>
<td>292</td>
</tr>
<tr>
<td>Get C IDENT</td>
<td>292</td>
</tr>
<tr>
<td>Get IDENT</td>
<td>292</td>
</tr>
<tr>
<td>Get IDENT NLS</td>
<td>293</td>
</tr>
<tr>
<td>Get IDENT NLS TRANS</td>
<td>293</td>
</tr>
<tr>
<td>Get PROJ_ID</td>
<td>294</td>
</tr>
<tr>
<td>IDENT_DESC</td>
<td>294</td>
</tr>
<tr>
<td>IDENT_DESC HIST</td>
<td>295</td>
</tr>
<tr>
<td>IDENT_LONG_DESC</td>
<td>295</td>
</tr>
<tr>
<td>New IDENT CODE</td>
<td>295</td>
</tr>
<tr>
<td>New RB IDENT CODE</td>
<td>295</td>
</tr>
<tr>
<td>NUMBER DN</td>
<td>296</td>
</tr>
<tr>
<td>NUMBER_SCHEDULE</td>
<td>296</td>
</tr>
<tr>
<td>POST_BUILD_IDENTS_CIP</td>
<td>297</td>
</tr>
<tr>
<td>Post_Final_Ids_CIP</td>
<td>297</td>
</tr>
<tr>
<td>S5007_Information</td>
<td>297</td>
</tr>
<tr>
<td>M_PCK_TAG_RUL_CUSTOM</td>
<td>298</td>
</tr>
<tr>
<td>Update CC Avec TAG DESC</td>
<td>298</td>
</tr>
<tr>
<td>UPD_TAGDESC_WITH_RLI_COMMENT</td>
<td>298</td>
</tr>
<tr>
<td>M_PCK_TRAFFIC_CUSTOM</td>
<td>298</td>
</tr>
<tr>
<td>Calc_sell_prices</td>
<td>299</td>
</tr>
<tr>
<td>ERP_TRANSFER</td>
<td>299</td>
</tr>
<tr>
<td>Get SUPP_LABEL</td>
<td>299</td>
</tr>
<tr>
<td>MOC_LABEL</td>
<td>300</td>
</tr>
<tr>
<td>Revert ERP_TRANSFER</td>
<td>300</td>
</tr>
<tr>
<td>M_PCK_UOM_CUSTOM</td>
<td>300</td>
</tr>
<tr>
<td>Get Value Ass</td>
<td>300</td>
</tr>
<tr>
<td>M_PCK_VDR_CUSTOM</td>
<td>301</td>
</tr>
<tr>
<td>Gen DOC NUMBER</td>
<td>301</td>
</tr>
<tr>
<td>Gen REFERENCE</td>
<td>301</td>
</tr>
<tr>
<td>Get ADDR TYPE</td>
<td>302</td>
</tr>
<tr>
<td>Get_CMT_EMAIL</td>
<td>302</td>
</tr>
</tbody>
</table>
Preface

This document describes the use of customer-installed procedures (CIPs) in SmartPlant Materials.

Documentation Comments

Send documentation comments or suggestions to PPMdoc@intergraph.com (mailto:ppmdoc@intergraph.com?subject=documentation comments).
Preface
SECTION 1

Overview

In many areas of SmartPlant Materials, you can adjust the standard functionality to match your company’s needs and requirements. These adjustments can be made by customizing the customer-installed procedures (CIPs).

This document describes all relevant CIPs with their default behavior and where they are used in the software.

IMPORTANT Customization should only be done by experienced developers or analysts who are familiar with SmartPlant Materials functionality and the underlying database structure.
To edit and compile the CIP procedures and functions, you can use:

- SQL*Plus (by Oracle) together with a text editor such as Notepad
- Any database development tool, for example, SQL Navigator (by Quest Software)
- A.60.37 Customer CIPs/Forms/Reports screen in SmartPlant Materials

The methods are described in the following chapters.

Although you can edit the package as a text file and compile it using the standard Oracle SQL*Plus tool, it is recommended that a development tool be used for ease of debugging.

For all three methods, the work must be done on a workstation with the Oracle client installed with configured SQLnet access to the SmartPlant Materials database.

**SQL*Plus**

By connecting to the database as M_SYS user, you can compile the modified package using SQL*Plus.

To spool out the package for editing, you must have two scripts: one for the package spec and one for the package body. The scripts start with the commands ‘CREATE OR REPLACE PACKAGE’ (for the spec) and ‘CREATE OR REPLACE PACKAGE BODY’.

With SQL*Plus, such a file can be created using the following statements (example from the M_PCK_VERIFY_CUSTOM package body):

```sql
set heading off
set feedback off
spool M_PCK_VERIFY_CUSTOM.PKB
select decode(line,1, 'CREATE OR REPLACE ' || text, text)
from    user_source
where   type = 'PACKAGE BODY'
and     upper(name) = 'M_PCK_VERIFY_CUSTOM'
order by line;
select '/' from dual;
spool off
```

This command spools the entire package body into a text file.

**NOTE** Before editing the custom package, you should save the original package code in a separate file.

Use an editor to open the M_PCK_VERIFY_CUSTOM.PKB file, apply your customizations, and save the modified file. Now run this file in SQL*Plus, which replaces the existing package in the database.

Finally, run the following command:

```sql
exec recomp;
```
Maintain Customer CIPs

to compile all objects that might have been invalidated by the changes.

**NOTE** During a migration or service pack installation, all custom packages are saved to files before the new version is installed. You can find these files in the `dbserver\scripts` directory. The SmartPlant Materials release notes provide information about which custom packages and which data objects have been changed.

**Development Tool**

Launch the development tool and open a session as M_SYS user. Select the custom package you want to edit and open the spec or body.

**NOTE** Before editing the custom package, you should save the original package code to a file.

When you finish editing the code, click **Save** to recompile.

**A.60.37 Customer CIPs/Forms/Reports**

You can use the **A.60.37 Customer CIPs/Forms/Reports** screen to view and maintain information about the use of customer-installed procedures and functions (CIPs) in forms, packages, stored procedures, and reports. Furthermore, you can analyze the packages to find related database tables, and you can edit and compile the CIP functions and procedures.

In the first (**CIPs**) block, all custom packages are listed with their descriptions.
Click [Analysse Package] to find the database tables that are used by the selected package. A window opens where the tables are listed.

In the second (CIP Refs) block, the functions and procedures that are included in the selected package appear. The **Object Name** field shows the name of the function or procedure, the **Object Type** field shows the type (FUNCTION or PROCEDURE), and the **Comment** field shows a brief description.
Maintain Customer CIPs

Click **Edit Object** to open Window 2.

In the first line, the **Object Name**, **Object Type**, and **Comment** fields appear.

The **Header** field shows the procedure/function header. The declaration section of the function/procedure is shown in the **Variables** field. The **Object text** field displays the executable section of the function/procedure.

Click **Compile** to compile the code of the function/procedure. The source code is not saved; it is only checked for inconsistencies. If there is an error, a warning message appears, and the error text appears in the field at the bottom of the screen.

Click **Update** to compile and save the changed function/procedure in the package. If there is an error, the source code is not saved, but a warning message appears, and the error text appears in the field at the bottom of the screen.

Click **Cancel** to cancel any changes and to return to the main screen.

In the third (**Menus**) block of the main screen, the screens that are linked to the selected CIP function/procedure appear.

At the top of the screen, you can see the **Create Delta File** and the **Display Version Delta** buttons. You can use these buttons to compare your database with a database that contains a newer version of SmartPlant Materials.

**NOTE:** To use this feature, you must create a database link to the remote database that has the newer SmartPlant Materials version installed.
Maintain Customer CIPs

Click to open Window 4.

Enter the user name for the login connection to the database with the newer version in the User Name field, with the password in the Password field. Enter the alias of the database with the newer version in the Database field.

The Path Name field displays the path on the database server where the delta file is going to be saved. This path is defined by the DBA setting CIPS_DELTA_PATH. Enter the name of the file with the database changes in the File Name field.

Click to create a delta file based on the information given. This delta file contains information about the database table changes from your version compared to the newer version.

Click to cancel the operation and return to the main screen.
Maintain Customer CIPs

On the main screen, click to open Window 5.

The Path Name field displays the path on the database server where the delta file is going to be saved. This path is defined by the DBA setting CIPS_DELTA_PATH. Enter the name of the file with the database changes in the File Name field.

Click to display the table changes between the current and the newer version.

Click to cancel the operation and return to the main screen.
**SECTION 3**

**Maintain CIP Relations**

You can use the **A.30.06 Menu Settings** screen to view and maintain the relationships between the SmartPlant Materials screens and the CIP procedures/functions.

On the **CIPs** tab of the A.30.06 screen, the CIP procedures and functions that are used by the selected screen (menu item) appear. The CIPs are shown with their name in the **Object Name** field, their type in the **Object Type** field, and a brief description of their purpose in the **Comment** field. The **Package Name** field displays the name of the custom package where you can find the CIP.
Maintain CIP Relations
View CIP Relations

From any screen in SmartPlant Materials, you can use the A.60.63 Valid Settings screen to view the CIP procedures/functions that are used on the screen. Click the Valid Settings button on the toolbar to open the A.60.63 screen and select the CIPs tab as shown in the picture below.

The name of the package is displayed in the Package Name field, the procedure or function name in the Object Name field, and the type in the Object Type field.
SECTION 5

CIP Example

The purpose of the SmartPlant Materials CIPs is explained in this chapter with an example procedure that can be used by a verification run on the B.20.01.41 Start Verification Job screen.

The CIP is intended to find some alternate idents for each BOM position where a pipe is located. The procedure looks for idents that can be used for the same specification, short code and sizes, but with different option codes. All alternate idents that are found are concatenated and written into a BOM position attribute. These alternate idents can be used later for reporting purposes. The reporting stage is not covered in this document.

Prerequisites

As a prerequisite, the BOM position attribute ALTERN_IDENT must be available. Create the ALTERN_IDENT attribute on the A.50.01 Attributes screen with Char= 60, Form Width=20, Unit= '-'. Next, assign this attribute to the PIPING discipline as a group dependent attribute for the piping commodity group on the B.10.04 Position Attributes screen and enable it by assigning it to a list view on the A.20.06.11 Attribute Security screen.

For more information, see Position Attributes and Discipline BOM Views in the SmartPlant Materials E&PI(BOM) User's Guide, available from the Printable Guides page in the software.

Create the CIP Procedure

First, add the following part to the package specification of m_pck_verify_custom:

```sql
PROCEDURE alte_ide
(p_lp_id IN m_list_pos.lp_id%TYPE);
-- This procedure finds alternate idents for a spec item
-- based on other option codes and put the list in
-- a position attribute ALTERN_IDENT of the BOM

The only parameter needed for this procedure is the BOM position unique key, m_list_pos.lp_id. The procedure is executed once for each position under the selected BOM node.

Next, add the second part of the code to the package body of m_pck_verify_custom:

```sql
PROCEDURE alte_ide
(p_lp_id IN m_list_pos.lp_id%TYPE)
IS
-- first declare variables to hold temporary results
spec_id m_list_pos.spec_header_id%type;
short_c m_list_pos.short_code%type;
help varchar(2000);
size1 m_list_pos.input_1%type;
alt_ident m_idents.ident%type;
option_pos m_list_pos.option_code%type;
intident m_idents.ident%type;
```
CURSOR optionc IS
SELECT option_code
FROM m_spec_items si
WHERE si.header_id = spec_id
    AND si.short_code = short_c
    AND size1 BETWEEN si.from_size1 AND si.to_size1
    AND si.option_code <> option_pos;
op_code optionc%rowtype;
BEGIN
    select lp.spec_header_id into spec_id
    from m_list_pos lp
    where lp.lp_id = p_lp_id;
    select option_code into option_pos
    from m_list_pos lp
    where lp.lp_id = p_lp_id;
    select short_code into short_c
    from m_list_pos lp
    where lp.lp_id = p_lp_id;
    select input_1 into size1
    from m_list_pos lp
    where lp.lp_id = p_lp_id;
    m_pck_m.ml('option_pos: ' || option_pos, 5, true);
    -- this procedure writes its first argument to a log file
    -- which is created on the DB server in the directory defined by
    -- the utl_file_dir parameter of the DBserver instance.
    -- It is here only for debugging purposes and could be
    -- commented out in a production environment
    m_pck_m.ml('short_c: ' || short_c, 5, true);
    m_pck_m.ml('p_lp_id: ' || p_lp_id, 5, true);
    -- select the possible option codes from the spec
    OPEN optionc;
    FETCH optionc INTO op_code;
    WHILE optionc%FOUND LOOP
        m_pck_m.ml('looping option_code: ' || op_code.option_code, 5, true);
        intident :=
        m_pck_idents.find_ident_org(spec_id, short_c, size1,'0','0',
        op_code.option_code,'0','0');
        -- this procedure fetches the ident from the spec
        -- it will return 0 if there is no corresponding ident
        m_pck_m.ml('intident: ' || to_char(intident), 5, true);
        if ( intident <> 0 ) then
            SELECT IDENT_CODE into help
            FROM M_IDENTS I WHERE I.IDENT = intident;
            -- this gives the ident code from the ident
            if (alt_ident is not NULL ) then
                SELECT IDENT_CODE into help
                FROM M_IDENTS I WHERE I.IDENT = intident;
                -- this gives the ident code from the ident
                if (alt_ident is not NULL ) then
                    alt_ident:=alt_ident||';'||help;
                    else
                        alt_ident:= help;
                    end if;
                -- this concatenates the ident codes
                end if;
                m_pck_m.ml('alt_ident: ' || alt_ident, 5, true);
            end if;
        end if;
    end loop;
END;
fetch optionc into op_code;
end loop;
close optionc;
m_pck_m.ml('value: ' || alt_ident, 5, true);
update m_list_pos_values lpv
set attr_value = alt_ident
where lpv.lp_id = p_lp_id
and lpv.attr_id IN (select attr_id from mvp_attrs a where a.attr_code = 'ALTERN_IDENT');
-- here the ALTERN_IDENT attribute is updated in BOM position
END alte_ide;

Now save and compile the package.

Use the Compiled CIP

Add a new list check of type PRC by selecting the new ALTE_IDE procedure from the LOV in the List Check field on the B.10.01 Verifications screen. If needed, double-click in the Where Condition field, and create a Where condition to be used for the check.

Now, open the B.20.01 Edit BOM screen, select a BOM node, and open the Single Position view. Select a position with piping material. Ensure that the ALTERN_IDENT position attribute is available. Click the button to open the B.20.01.41 Start Verification Job screen. Create a list job, select a list status from the LOV, and select the Best Qty option in the Issue Status Selection section. In the List Job Checks section, select the new ALTE_IDE from the LOV in the List Check field. Save and click Start to run the verification job.

Close B.20.01.44 to return to the Single Position view. Re-query the data to refresh the screen. The Value field of the ALTERN_IDENT attribute shows the appropriate ident codes, separated by semi-colons.
SECTION 6

Custom Packages

This section provides an overview about the available custom packages in SmartPlant Materials. For each package, the functions and procedures are listed with:

- a brief description of their purpose
- the screen from which they are called
- the event that triggers their execution
- the default behavior, that is, how they behave without any customization

The packages and the functions and procedures within the packages are listed in alphabetical order.

M_PCK_ACCOUNT_CUSTOM

This package contains CIP procedures for the handling of account codes.

ACCOUNT_QUERY

Description: Data source for the Account Summary tab on R.30.01 and P.50.07.

Type: Procedure

Usage: P.50.07, R.30.01

Event: Executed by CHECK_BUDGET_PO and CHECK_BUDGET_REQ

Default: The procedure calculates the budget and original budget for each account code.

**CALC_DISPO**

Description: This procedure is used to calculate the dispo quantity.

Type: Procedure

Usage: R.30.01

Event: Executed by CALC_RELEASE

Default: Calculates the delta quantity between the recommended quantity and the BOM quantity of a requisition line item and assigns this delta (if <> 0) to the account code defined by project default ZJ_AC_OVER.

**CALC_ORDER_ACCOUNTS**

Description: This procedure calculates the overage order accounts.

Type: Procedure

Usage: P.30.01, P.30.02, P.30.04, P.50.01, P.50.07

Event: Creation or approval of an agreement

Default: Compares the quantity and budget of an agreement line item with the current quantity and budget on order accounts and assigns delta values to the account code defined by project default ZJ_AC_OVER.

**CALC_RELEASE**

Description: This procedure calculates the overage requisition accounts.

Type: Procedure

Usage: R.30.01

Event: None

Default: Calculates the delta quantity between the recommended quantity and the released quantity of a requisition line item and assigns this delta (if <> 0) to the account code defined by project default ZJ_AC_OVER.
CHECK_BUDGET_PO

Description: If the project default ZJ_ACC_CHK is set to Y, this function checks whether approval of an agreement is allowed.

Type: Function

Usage: P.50.07

Event: Agreement approval

Default: If budget exceeded -> FALSE. Else -> TRUE

CHECK_BUDGET_REQ

Description: If the project default ZJ_ACC_CHK is set to Y, this function checks whether RTP (release to procurement) for a requisition is allowed.

Type: Function

Usage: R.30.01

Event: Release to procurement of the requisition (click Release on R.30.01)

Default: If budget exceeded -> FALSE. Else -> TRUE

DISTRIB_OTHER_COSTS

Description: This procedure is used to distribute other costs on the line item level over the assigned account codes.

IMPORTANT This procedure must not commit.

Type: Procedure

Usage: P.30.01, P.30.02, P.30.03, P.30.04, P.30.05, P.30.21, P.30.22, P.30.27, P.50.07, R.30.01

Event: Whenever other costs are automatically assigned to quote details or agreement line items and when assigning other costs to requisition line items on the R.30.01 screen.

Default: NULL
**FETCH_PROGRESS**

**Description:** This function is used for calculating the Progress in % value on the J.20.11 Export Order Progress screen. The function is called once for every selected order.

**Type:** Function

**Usage:** J.20.11

**Event:** Export (click Export on J.20.11)

**Default:** 100%

---

**GET_BUDGET_DATA**

**Description:** This procedure calculates the value of the Forecast, Actual PO, Actual Req, and Remaining Budget fields on A.10.30/J.10.03 Account Codes. Depending on the setting of project default ZJ_CAL_MOD, the calculation is done:

- **H** - on the header level
- **I** - on the line item level
- **M** - the maximum of both values is used
- **PH** - priority is given to the header level calculation when the other value is empty
- **PI** – priority is given to the line item level calculation when the other value is empty

**Type:** Procedure

**Usage:** A.10.30, J.10.03

**Event:** Open screen or query data on the screen

**Default:** See CIP source code
M_PCK_APPROVAL_CUSTOM
This package contains custom functions and procedures for web approval.

BUILD_APPROVAL_BODY
Description: This function builds the text in the body of the approval e-mail message.
Type: Function
Usage: A.20.49, Web Approval
Event: Click Approve button
Default: This is to inform you that the following <object type> is ready for approval.’
(<object_type> is a placeholder for the type of the object to be approved),
followed by the object details.

GET_APPROVAL_TEXT1
Description: This function delivers the text for line 1 on the web approval screen
(WWW_APPROVE.FMB).
Type: Function
Usage: Web Approval
Event: Open the web approval screen
Custom Packages

Default: The returned text depends on the object to be approved. The type of the object is controlled by p_pk_type. The available values of p_pk_type are listed below.

<table>
<thead>
<tr>
<th>pk_type</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>Requisition</td>
</tr>
<tr>
<td>LR</td>
<td>Login Request</td>
</tr>
<tr>
<td>CP</td>
<td>Company Profile</td>
</tr>
<tr>
<td>Q</td>
<td>Qualification</td>
</tr>
<tr>
<td>INQ</td>
<td>Inquiry</td>
</tr>
<tr>
<td>SUP</td>
<td>Inquiry Bidders (supplier)</td>
</tr>
<tr>
<td>ADD</td>
<td>Inquiry Addendum</td>
</tr>
<tr>
<td>QS</td>
<td>Quote Summary</td>
</tr>
<tr>
<td>PO</td>
<td>Purchase Order</td>
</tr>
<tr>
<td>NC</td>
<td>Notice of Commitment</td>
</tr>
<tr>
<td>BO</td>
<td>Blanket Order</td>
</tr>
<tr>
<td>SC</td>
<td>Subcontract</td>
</tr>
<tr>
<td>PNO</td>
<td>Period Progress Number</td>
</tr>
<tr>
<td>MSP</td>
<td>Milestone Progress</td>
</tr>
<tr>
<td>SOV</td>
<td>Schedule of Values</td>
</tr>
<tr>
<td>SOVP</td>
<td>Schedule of Values Progress</td>
</tr>
<tr>
<td>COR</td>
<td>Change Order Request</td>
</tr>
<tr>
<td>INV</td>
<td>Invoice</td>
</tr>
<tr>
<td>BC</td>
<td>Backcharge</td>
</tr>
<tr>
<td>NCR</td>
<td>Non conformance Report</td>
</tr>
<tr>
<td>CLOI</td>
<td>Closeout Checklist Item</td>
</tr>
<tr>
<td>MIR</td>
<td>Material Issue Report</td>
</tr>
</tbody>
</table>

For example, the following text would be returned for a quote summary (QS):
The following quote summary is ready for approval:

Code: <inq_code>, Suppl: <i_supp>, Purchase: <actual_ind>

The picture below shows the web approval screen with the customizable text lines.
**GET_APPROVAL_TEXT2**

Description: This function delivers the text for line 2 on the web approval screen (WWW_APPROVE.FMB).

Type: Function

Usage: Web Approval

Event: Open the web approval screen

Default: ‘Please approve or reject’

**GET_APPROVAL_TEXT3**

Description: This function delivers the text for line 3 on the web approval screen (WWW_APPROVE.FMB).

Type: Function

Usage: Web Approval

Event: Open the web approval screen

Default: By clicking Approve/Reject I certify that I am #1 or I’ (#1 is a placeholder for the approver name)

**GET_APPROVAL_TEXT4**

Description: This function delivers the text for line 4 on the web approval screen (WWW_APPROVE.FMB).

Type: Function

Usage: Web Approval

Event: Open the web approval screen

Default: ‘have the authority to take this action.’
M_PCK_BC_CUSTOM

This package contains all customer-defined functions for the BOM compare functionality.

BOM_DISPO_QTY

Description: This function calculates the dispo quantities based on the requisition dispo details.

Type: Function

Usage: B.40.22

Event: Start button on B.40.22 (m_pck_bom_compare.start_job)

Default: See source code

GET_ER_QTY_TOTAL

Description: This sample function can be used for testing the BOM compare functionality.

Type: Function

Usage: B.40.22

Event: Start button on B.40.22

Default: sum(a.total_release_qty - a.last_total_release_qty) from mvp_req_line_items

GET_ONLY_PO_QTY_TOTAL

Description: This sample function can be used for testing the BOM compare functionality.

Type: Function

Usage: B.40.22

Event: Start button on B.40.22

Default: sum(po_qty) from mvp_req_line_items
**GET_PO_OR_ER_QTY_TOTAL**

Description: This sample function can be used for testing the BOM compare functionality.

Type: Function

Usage: B.40.22

Event: **Start** button on B.40.22

Default: \[\text{sum}(\text{decode(a.po_qty,0,(a.total_release_qty - a.last_total_release_qty),0)}), \text{sum}(\text{decode(a.po_qty,0,0,a.po_qty)})\]

**SET_QUANTITY**

Description: This function can be assigned to a BOM compare structure or to BOM compare structure results on the B.40.21 BOM Compare Structures screen. You can select the CIP from the LOV in the CIP Name fields. This function updates the quantities in the BOM_COMPARE_RESULTS table.

Type: Function

Usage: B.40.22

Event: **Start** button on B.40.22

Default: 0

**SET_SELECT**

Description: This function sets the Select indicator on the B.40.23 Compare Results screen (table M_BOM_COMP_RESULTS). This CIP function is called for each ident, so the Select indicator can be used to show if the CIP function has processed/changed this record or not.

Type: Function

Usage: B.40.22

Event: **Start** button on B.40.22

Default: Y when qty_1, 2, 3, 4, 5, 6, or 7 < 0, else N
Custom Packages

**M_PCK_BIR_CUSTOM**

This package contains all the custom procedures and functions that are used in SmartPlant Materials Business Intelligence Reporting.

**GET_ACTUAL_SHIP_DATE**

Description: This function returns the ACTUAL_CONTRACT_DATE from M_ITEM_SHIPS by default. You can use other logic, if needed.

Type: Function

Usage: BIR: Adherence to Delivery Date report

Event: Called from BIR page 180 - Adherence to Delivery Date

Default: m_sys.m_item_ships.actual_contract_date

**GET_BIR_CONVERTED_VALUE**

Description: This function returns a modified value for Business Intelligence Reporting in order to display SmartPlant Materials data correctly in the reports. For example, you can remove linefeeds or replace double quotes with single quotes.

After each change of this CIP function, the BOM ident data must be re-created. To do this, proceed as follows:

1. Launch a SQL*Plus session, and connect to the Business Intelligence Reporting - SmartPlant Materials schema.

2. Clear the bir_spmat_identdata table:
   ```sql
   truncate bir_spmat_identdata;
   ```

3. Call the sync data script:
   ```sql
   begin
   bir_bi_functions.sync_data();
   end;
   ```

Type: Function

Usage: BIR

Event: Data synchronization

Default: replace(replace(p_invalue,chr(10),' '),chr(34),chr(39)||chr(39))
GET_PLANNED_SHIP_DATE

Description: This function returns the PROM_CONTRACT_DATE from M_ITEM_SHIPS by default. You can use other logic, if needed.

Type: Function

Usage: BIR: Adherence to Delivery Date report

Event: Called from BIR page 180 - Adherence to Delivery Date

Default: m_sys.m_item_ships.prom_contract_date

GET_PSR_CIP1 to _CIP10

Description: This function is called from page P190 - Procurement Status report. It returns a string to be shown in the report. Possible inputs are:

- Req_id: Requisition ID
- Ident: Ident number
- NLS_id: Language ID
- POH_ID: Agreement / purchase order ID

If the value does not exist, the function returns NULL.

Type: Function

Usage: BIR: Procurement Status report

Event: Called from BIR page 190 - Procurement Status

Default: m_list_node_values.attr_value for AREA CODE attribute
**GET_PSR_CIP2**

**Description:** This function is called from page P190 - Procurement Status report. It returns a string to be shown in the report. Possible inputs are:

- **Req_id:** Requisition ID
- **Ident:** Ident number
- **NLS_id:** Language ID
- **POH_ID:** Agreement / purchase order ID

If the value does not exist, the function returns NULL.

**Type:** Function

**Usage:** BIR: Procurement Status report

**Event:** Called from BIR page 190 - Procurement Status

**Default:** `m_list_node_values.attr_value` for `REQ_ATTRI_1` attribute

**GET_PSR_CIP3 to _CIP10**

**Description:** This function is called from page P190 - Procurement Status report. It returns a string to be shown in the report. Possible inputs are:

- **Req_id:** Requisition ID
- **Ident:** Ident number
- **NLS_id:** Language ID
- **POH_ID:** Agreement / purchase order ID

If the value does not exist, the function returns NULL.

**Type:** Function

**Usage:** BIR: Procurement Status report

**Event:** Called from BIR page 190 - Procurement Status

**Default:** "
GET_PSR_CIP1_HEADER

Description: This function is called from page P190 - Procurement Status report. It returns the column header for CIP1 for the interactive report.

Type: Function

Usage: BIR: Procurement Status report

Event: Called from BIR page 190 - Procurement Status

Default: 'Area'

GET_PSR_CIP2_HEADER

Description: This function is called from page P190 - Procurement Status report. It returns the column header for CIP2 for the interactive report.

Type: Function

Usage: BIR: Procurement Status report

Event: Called from BIR page 190 - Procurement Status

Default: 'Req Attribute'

GET_PSR_CIP3_HEADER to _CIP10_HEADER

Description: This function is called from page P190 - Procurement Status report. It returns the column header for CIP3...10 for the interactive report.

Type: Function

Usage: BIR: Procurement Status report

Event: Called from BIR page 190 - Procurement Status

Default: '-na-'
M_PCK_BOMCUSTOM

This package contains all the procedures and functions of the BOM that can be customized.

GET_LP_REQ_IND

Description: This CIP function allows you to define whenever the requisition flag (requisition_ind) should be set in the m_list_pos table after an insert or update operation. The function is called within a trigger, so don't use commit in the function body. The function must return Y when the requisition flag should be set, or N when the requisition flag should not be set. Use '-' as the return value when the requisition flag should not be changed.

Type: Function
Usage: B.20.01, B.20.02, B.20.03, B.40.01
Event: Called by m_pck_lp.update_lp_req_ind.
Default: Y when qty_1, 2, 3, 4, 5, 6, or 7 < 0, else N

M_PCK_COMPANYCUSTOM

This package contains all procedures and functions for companies and suppliers as a special occurrence of companies that can be customized.

CHANGED_PW_USER_EMAIL_TEXT

Description: CIP for the e-mail text for a changed user password.

Type: Function
Usage: P.20.02
Event: Change user password
Default: 'Your user ID for supplier login for company <company_code> is <client_user_id>.'

'Your password changed to: <password>'

<company_code> is replaced with the addressee's company, <client_user_id> with the username for the E-Supplier login, and <password> with the user password.
CHECK BEFORE SUBMIT

Description: This CIP is executed when the Submit button is clicked on the Summary page in SmartPlant Materials Portal or on the P.20.02 Login Requests screen. You can use this CIP to implement checks or actions that must be performed before the login request is submitted. In the case of an error, the data already entered and saved is deleted.

Type: Procedure

Usage: SmartPlant Materials Portal, Company Registration - Login Request

Event: Click Submit button

Default: NULL

CHECK COMPANY_DATA1

Description: This CIP checks that the combination of company name, country, and postal code is unique. If data exists, p_exists_ind is set to Y and an appropriate MAR message is displayed.

Type: Procedure

Usage: SmartPlant Materials Portal, Company Registration

Event: Insert or update company data

Default: Null if p_exists_ind=N; MAR message if p_exists_ind=Y

CHECK COMPANY_DATA2

Description: This CIP checks that the combination of company name and federal ID (tax ID, business registration number) is unique. If data exists, p_exists_ind is set to Y and an appropriate MAR message is displayed.

Type: Procedure

Usage: SmartPlant Materials Portal, Company Registration

Event: Insert or update company data

Default: Null if p_exists_ind=N; MAR message if p_exists_ind=Y
CHECK.Company_Profile_Approval

Description: CIP for check company profile approval
Type: Function
Usage: P.20.04
Event: Open screen
Default: TRUE

CHECK.ESUPP.For.Branch

Description: This CIP checks whether the Branch Out To button can be activated on the E-Supplier logon screen and on the SmartPlant Materials Portal logon screen. The parameter is the supplier ID sup_id.
If the return value is Y, the button is enabled; otherwise not.
Type: Function
Usage: SmartPlant Materials Portal home page, P.20.04
Event: Open screen
Default: Y

CHECK.Insert.Company

Description: CIP called by a screen trigger after a company was inserted
Type: Function
Usage: A.10.23, P.20.01
Event: Insert new company
Default: TRUE
CHECK_INSERT_SUPPLIER
Description: CIP called by screen trigger after a supplier was inserted
Type: Function
Usage: P.20.01
Event: Insert new supplier
Default: TRUE

CHECK_LOGIN_APPROVAL
Description: CIP for check login approval
Type: Function
Usage: P.20.02
Event: Open screen
Default: TRUE

CHECK_UPDATE_COMPANY
Description: CIP called by a screen trigger after a company is updated
Type: Function
Usage: A.10.23
Event: Update company
Default: TRUE
Custom Packages

CHECK_UPDATE_SUPPLIER
Description: CIP called by a screen trigger after a supplier is updated.
Type: Function
Usage: P.20.01
Event: Update supplier
Default: TRUE

COMPANY_APPROVAL_EMAIL_TEXT
Description: CIP for the e-mail text for the approval of a login request.
Type: Function
Usage: P.20.02
Event: Login request approval
Default: 'Dear <name>,'
'Your login request of company <company_code> - <company_name>, type <company_type> has been accepted.'
'To continue the process for accepted Login Requests please return to the system and provide your Company Profile.'
'This e-mail has been generated automatically. Please do not reply.'
(<name> is replaced with the username, <company_code> with the company code, <company_name> with the name of the company, and <company_type> with the type, for example, SUPPLIER)
CP_PROFILE_APPROVAL_TEXT

Description: CIP for the e-mail text for the approval of a company profile. Return NULL if you want to use the standard functionality.

Type: Function

Usage: A.20.49

Event: Approval or rejection of company profile

Default: 'Information about status change of company profile:'
'Acepted by: <first_name> <last_name>,'
'Comment on the decision: <comment>,'
'Please login to enter your qualification !'

(<first_name> and <last_name> are replaced by the first and last name of the approver; <comment> is replaced by the comment given at the company profile approval/rejection.)

GET_CLIENT_USER_ID

Description: CIP for setting the client user ID of login request contacts

Type: Function

Usage: P.20.02

Event: Login request approval

Default: m_login_request_contacts.ccp_code

GET_COMPANY_CITY

Description: This CIP returns the company city including the state and postal code. The parameter is the company address ID CA_ID.

Type: Function

Usage: P.20.02

Event:

Default: City, state, postal code
### GET_COMPANY_CODE

**Description:** This CIP returns a unique company code based on the company name entered during registration.

**Type:** Function

**Usage:** SmartPlant Materials Portal, Company Registration

**Event:** Enter company name

**Default:** `company_code`

### GET_FIRST_AWARDING_FIELD

**Description:** CIP for setting the value for the first matrix header field for awarding (*Awarding* tab on the [P.30.22 Commercial Evaluation](#) screen)

**Type:** Function

**Usage:** P.30.25

**Event:** Open screen

**Default:** `Supplier name / Buyer` (name is repeated in the header if project default `ZP_CE_RPV` is set to Y)

### GET_FIRST_AWARDING_PROMPT

**Description:** CIP for setting the prompt of the first matrix header field for awarding (*Awarding* tab on the [P.30.22 Commercial Evaluation](#) screen)

**Type:** Function

**Usage:** P.30.25

**Event:** Open screen

**Default:** `'Bidder / Buyer'`
Custom Packages

**GET_FIRST_FIELD**
Description: CIP for retrieving the value for the first matrix header field
Type: Function
Usage: P.30.27
Event: Open screen
Default: m_suppliers.sup_code (supplier code)

**GET_FIRST_PROMPT**
Description: CIP for setting the prompt of the first matrix header field
Type: Function
Usage: P.30.22, P.30.23, P.30.25, P.30.27
Event: Open screen
Default: 'Bidder'

**GET FOURTH FIELD**
Description: CIP for retrieving the value for the fourth matrix header field
Type: Function
Usage: P.30.22, P.30.23, P.30.25
Event: Open screen
Default: m_inq_suppliers.buyer
Custom Packages

GET_FOURTH_PROMPT

Description: CIP for setting the prompt of the fourth matrix header field
Type: Function
Usage: P.30.22, P.30.23, P.30.25
Event: Open screen
Default: 'Buyer'

GET_REPORT_URLS

Description: This CIP is called by SmartPlant Materials Portal upon certain context changes. The context parameters are passed to the CIP to determine which reports are available in a given context. The reports (title/URL) are returned in a context-specific collection of customer reports that are accessible to the end user in SmartPlant Materials Portal via View Reports.
Type: Function
Usage: SmartPlant Materials Portal
Event: Login to Portal
Default: Sample statement available

GET_SECOND_FIELD

Description: CIP for retrieving the value for the second matrix header field
Type: Function
Usage: P.30.27
Event: Open screen
Default: m_companies.company_name
GET_SECOND_PROMPT
Description: CIP for setting the prompt of the second matrix header field
Type: Function
Usage: P.30.22, P.30.23, P.30.25, P.30.27
Event: Open screen
Default: 'Company Name'

GET_SUPPLIER_STATUS
Description: This CIP returns the supplier status. The parameter is the supplier ID sup_id.
Type: Function
Usage:
Event:
Default: 'Approved' if approved_ind=Y; else 'Active'.

GET_THIRD_FIELD
Description: CIP for retrieving the value for the third matrix header field
Type: Function
Usage: P.30.27
Event: Open screen
Default: m_companies.company_code
Custom Packages

**GET_THIRD_PROMPT**

Description: CIP for setting the prompt of the third matrix header field

Type: Function

Usage: P.30.22, P.30.23, P.30.25, P.30.27

Event: Open screen

Default: ‘Company’

**INS_COMPANY_CIP**

Description: CIP called by a database trigger after a new company is inserted

Type: Procedure

Usage: A.1023, A.20.49, OSCI.10.01

Event: Insert a new company

Default: NULL

**INS_SUPPLIER_CIP**

Description: CIP called by a database trigger after a supplier is inserted

Type: Procedure

Usage: A.10.23, A.10.24, A.20.49, OSCI.10.01, P.20.01

Event: Insert a new supplier

Default: NULL
LOGIN_REQUEST_INFORMATION_TEXT

Description: CIP for the e-mail text for the approval of a login request
Type: Function
Usage: P.20.02, P.20.03
Event: Login request approval
Default:
'Type        :<ct_code>'
'Address     :<addr1>'
   '<city>'
'          <state>, <postal_code>'
'           <cy_description> (or <cy_code>)'
(the variables in <> are replaced by the appropriate values)

NEW_USER_EMAIL_TEXT

Description: CIP for the e-mail text for a new user
Type: Function
Usage: P.20.02, E-Supplier portal, Web Approval
Event: Create new user
Default:
'Your user ID for supplier login for company <company_code> is
   <client_user_id>, Password: <password>'
(the variables in <> are replaced by the appropriate values)
**SUPPLIER_PASSWORD_TEXT**

Description: CIP for the e-mail text for a supplier password request

Type: Function

Usage: E-Supplier portal

Event: Click *Forgot Password* button

Default: 'Your user ID for supplier login for supplier *sup_code* is *user_id*, Password: *password*'

**UPD_COMPANY_CIP**

Description: CIP called by a database trigger after a company is updated

Type: Procedure

Usage: A.10.23, A.20.49, OSCI.10.01

Event: Update a company

Default: NULL

**UPD_SUPPLIER_CIP**

Description: CIP called by a database trigger after a supplier is updated

Type: Procedure

Usage: A.10.23, A.10.24, A.20.49, OSCI.10.01, P.20.01

Event: Update a supplier

Default: NULL
M_PCK_CONFIG_CUSTOM

This package contains all procedures and functions for configurable screens.

GET_CIP_LABEL

Description: Function for returning the label for a CIP
Type: Function
Usage: D.10.11, D.40.12, D.90.21, P.20.01, P.30.01, P.30.11, P.30.22, P.30.23, P.30.24, P.30.25, P.30.27, P.30.81, P.50.01, P.50.07, P.50.81, P.60.01, P.60.03, P.60.11, P.70.71, P.70.72, P.70.73, P.70.74, P.70.81, P.80.42, P.90.11, PT.20.12, R.30.01, SM.20.01, SM.20.11
Event: Open screen
Default: NULL
If m_config_details.cip_name=GET_COMPANY_NAME -> ‘Company Name’
If m_config_details.cip_name=GET_CYCLE_APPROVED_IND -> ‘Approval Status’

GET_COMPANY_NAME

Description: Example of a function for suppliers
Type: Function
Usage: D.40.12, D.90.21, P.20.01, P.30.11, P.30.22, P.30.23, P.30.24, P.30.25, P.30.27, P.30.81, P.50.01, P.50.07, P.50.81, P.60.01, P.60.03, P.60.11, P.70.71, P.70.72, P.70.74, P.80.42, PT.20.11, R.30.01, SM.20.01, SM.20.11
Event: Open screen
Default: m_companies.company_name
Custom Packages

GET_CYCLE_APPROVED_IND

Description: Example of a function for returning an indicator for whether orders not approved so far exist for the given order cycle
Type: Function
Usage: P.50.01
Event: Open screen
Default: ‘Approved’ or ‘Open’

M_PCK_CUSTOM_ATTRS

This package contains customizable lookup functions for ident attribute values.

IDENT_CODE

Description: This function returns the ident code for a given ident.
Type: Function
Usage: A.60.14
Event:
Default: mvp_idents.ident_code

IDENT_DESC

Description: This function returns the language dependent description for a given ident.
Type: Function
Usage: A.60.14
Event:
Default: m_pck_std_custom.ident_desc(p_ident, p_nls_id)
This package contains all CIP functions that may be part of document type rules.

**CIP1**

**Description:** This function is intended to return the unit for a requisition header/requisition line. The function can be assigned to a document type rule on the **R.10.24 Document Types** screen.

**Type:** Function

**Usage:** R.30.01

**Event:**

**Default:** 'CIP1_REQHEADER' for requisition header or 'CIP1_REQ_LINE' for requisition line

**CIP2**

**Description:** This function is intended to return the material code for a requisition line. The function can be assigned to a document type rule on the **R.10.24 Document Types** screen.

**Type:** Function

**Usage:** R.30.01

**Event:**

**Default:** 'CIP2_REQHEADER' for requisition header or 'CIP2_REQ_LINE' for requisition line
**CIP3**

Description: This function is intended to be implemented by the customer. The function can be assigned to a document type rule on the **R.10.24 Document Types** screen.

Type: Function

Usage: R.30.01

Event:

Default: 'CIP3_REQHEADER' for requisition header or 'CIP3_REQ_LINE' for requisition line

---

**GET_TABLE_DETAIL_INFO**

Description: This procedure gets the table detail information from the document.

Type: Procedure

Usage: SmartPlant Reference Data Plus

Event:

Default: Table detail with short description and description

---

**M_PCK_EMAIL_CUSTOM**

This package provides utilities to send Req to Proc e-mail messages called by M_PCK_DOC_INTERFACES PDB / VDB generation storing and sending.

---

**BUILD_DOCUMENT_REFERENCE**

Description: Example function to build a document reference

Type: Function

Usage: R.30.01

Event:

Default: mpck_login.proj_id|| ' SR ' || R_CODE || ' ' || ACTUAL_IND || ' ' || ORIGIN_ID
**EMAIL_BODY_TEMPLATE**

Description: Return examples of the e-mail text with all possible substitution variables

Type: Function

Usage: A.10.51

Event:

Default: Sample e-mail texts for PSW USR CHANGED, REQUISITION, FSMN

**EMAIL_EXAMPLE**

Description: Example function to fill an e-mail body and send the e-mail message

Type: Function

Usage: R.30.01

Event:

Default: Sample for e-mail body generation

**EMAIL_SUBJ_TEMPLATE**

Description: Return example of the e-mail subject with all possible substitution variables if any

Type: Function

Usage: A.10.51

Event:

Default: Sample e-mail subjects for
PSW USR CHANGED: 'Your account is changed.'
REQUISITION: NULL
FSMN: NULL
Custom Packages

FILENAME_INIT
Description: Procedure to initialize the PL-SQL table filename_tab of filename_tab_type
Type: Procedure
Usage: R.30.01
Event: zipcode_tab.DELETE

GET_SUBJECT_BODY_ANSWER
Description: This procedure returns the subject and body for the email message that is sent to users when they forget the answer to their security question.
Type: Procedure
Usage: eSignature
Event: Click Forgot your answer link, or type an incorrect answer three times.
Default: subject: "Incorrect Answer Provided"
body: "The answer you have provided has been typed in incorrectly 3 times. The correct answer is : " || p_answer;
or
subject: "No Answer Provided"
body: "No answer has been provided for the question"

OTHER_REPORT_PROP
Description: Function to return the report properties
Type: Function
Usage: R.30.01
Event:
' copies=1'
' readonly=yes'
' paramform=no'
' background=no'
' p_r_id='||p_r_id
' p_us_id='||mpck_login.us_id

PDF_FILE_NAME
Description: Function to build a new unique file name for the pdf file and save it to the PL-SQL table
Type: Function
Usage: R.30.01
Event:
Default: Sample for pds-filename generation

REPORT EXAMPLE
Description: Example function to return a report name
Type: Function
Usage: R.30.01
Event:
Default: 'requisition.rdf'
Custom Packages

REPORT_EXAMPLE2
Description: Example function to return a report name
Type: Function
Usage: R.30.01
Event:
Default: 'requisition2.rdf'

REPORT_EXAMPLE3
Description: Example function to return a report name
Type: Function
Usage: R.30.01
Event:
Default: 'requisition3.rdf'

REPORT_EXAMPLE4
Description: Example function to return a report name
Type: Function
Usage: R.30.01
Event:
Default: 'requisition4.rdf'
ZIPNAME_INIT

Description: Procedure to initialize the PL-SQL table zipname_tab of zip_tab_type

Type: Procedure

Usage: R.30.01

Event:

Default: zipname_tab.DELETE

ZIP_FILE_NAME

Description: Function to build a new unique zip name for the attachments and save it to the PL-SQL table

Type: Function

Usage: R.30.01

Event:

Default: Sample for zip file name generation

M_PCK_EXP_CUSTOM

This package contains all procedures and functions that can be customized for expediting.

CHECK_CHANGE_SG

Description: This procedure is called before shipment groups are changed. If this procedure yields any error, shipment groups are NOT modified.

Type: Procedure

Usage: C.20.01, C.20.01.02, C.20.04, C.20.05, OPI.10.01, P.50.07, P.70.31, P.70.62, P.70.72, P.70.74, P.80.11, P.80.51

Event: Change shipment group

Default: NULL
CHECK_CREATE_ISH

Description: This procedure is called when you click the Create Item Shipment button on the P.70.72 Expediting Details screen. If this procedure yields any error, the item shipment is NOT created.

Type: Procedure
Usage: P.70.72
Event: Create item shipment
Default: NULL

CHECK_CREATE_SG

Description: This procedure is called before shipment groups are created. If this procedure yields any error, shipment groups are NOT created.

Type: Procedure
Usage: P.70.31, P.70.61
Event: Create shipment group
Default: NULL

CHECK_DELETE_SG

Description: This procedure is called before shipment groups are deleted. If this procedure yields any error, shipment groups are NOT deleted.

Type: Procedure
Usage: P.70.31
Event: Delete shipment group
Default: NULL
CHECK_SPLIT_TAG
Description: This procedure is called before tags are split. If this procedure yields any error, tags are NOT split.
Type: Procedure
Usage: C.20.01, C.20.01.02, C.20.04, C.20.05, OPI.10.01, P.30.22, P.30.27, P.50.07, P.70.31, P.70.62, P.70.72, P.70.74, P.80.11, P.80.51
Event: Split Tag
Default: NULL

GET_EXPED_STATUS
Description: This function can be used, for example, by m_pck_config_custom to return the expediting status including color and priority information of an agreement, a shipment group, or a release note and display this information in a CIP field on the P.70.71 Expediter Workload or the P.70.72 Expediting Details screens.
Type: Function
Usage:
Event:
Default: Status, color, priority

POST_IE_ATTR
Description: After an attribute has been changed on the Inspection Events tab on the P.70.73 Order Reference screen, this CIP is executed.
   The parameter of the CIP is the primary key uval_id of the m_used_values table.
Type: Procedure
Usage: P.70.73
Event: Modify inspection event attribute
Default: NULL
**POST_RELN_APPROVAL**

Description: This procedure is executed after a release note has been approved.

Type: Procedure

Usage: P.70.72

Event: Approve release note

Default: NULL

**POST_RELN_REV_APPROVAL**

Description: This procedure is executed after the approval of a release note has been reversed (reject approval).

Type: Procedure

Usage: P.70.72

Event: Reject release note approval

Default: NULL

**POST_SPLIT_TAG**

Description: This procedure is called after a detail tag has been created and all data has been committed. You must include an additional COMMIT statement if you modify any data in this procedure. This CIP can be used, for example, to update the description of the new detail tag.

- pk_type = 'QD' -> pk_id contains qd_id coming from m_quote_details
- pk_type = 'POLI' -> pk_id contains poli_id coming from m_po_line_items
- pk_type = 'ISH' -> pk_id contains item_ship_id coming from m_item_ships

Type: Procedure

Usage: C.20.01, C.20.01.02, C.20.04, C.20.05, OPI.10.01, P.30.27, P.50.07, P.70.31, P.70.62, P.70.72, P.70.74, P.80.11, P.80.51

Event: Split Tag
Default: NULL (sample code available)

**PRE_RELN_APPROVAL**

Description: This procedure is executed before a release note is approved.

Type: Procedure

Usage: P.70.72

Event: Approve release note

Default: NULL

**PRE_RELN_REV_APPROVAL**

Description: This procedure is executed before the approval of a release note is reversed (reject approval).

Type: Procedure

Usage: P.70.72

Event: Reject release note approval

Default: NULL

**M_PCK_FAB_CUSTOM**

This package contains all procedures and functions that can be customized for fabrication tracking.

**POST_IMPORT**

Description: This procedure is executed at the end of fab tracking import on the P.90.01 Load Tab Tracking screen.

Type: Procedure

Usage: P.90.01
Custom Packages

Event: Import fab tracking
Default: NULL

**POST_LOAD**

Description: This procedure is executed at the end of a fab tracking load on the P.90.11 Maintain Fab Tracking screen.

Type: Procedure
Usage: P.90.11
Event: Load fab tracking
Default: NULL

**PRE_IMPORT**

Description: This procedure is executed at the beginning of fab tracking import on the P.90.01 Load Tab Tracking screen.

Type: Procedure
Usage: P.90.01
Event: Import fab tracking
Default: NULL

**PRE_LOAD**

Description: This procedure is executed at the beginning of a fab tracking load on the P.90.11 Maintain Fab Tracking screen.

Type: Procedure
Usage: P.90.11
Event: Load fab tracking
Default: NULL
**UPDATE_FT**

**Description:** This procedure is executed when you click the **Execute CIP** button on the **P.90.11 Maintain Fab Tracking** screen.

**Type:** Procedure

**Usage:** P.90.11

**Event:** Click **Execute CIP** button

**Default:** NULL

---

**M_PCK_FORECAST_CUSTOM**

This package contains all customer-modifiable logic related to C.30.xx.

---

**CHECK_BEFORE_RR**

**Description:** This check procedure is executed before a forecast run (C.30.01) or a reservation run (C.30.11) starts. The procedure should be modified according to the customer's business requirements. The procedure has two output parameters:

- **return_number:**
  - 0 nothing to do
  - 1 alert will inform user

- **return_value:** alert text

**Type:** Procedure

**Usage:** C.30.01, C.30.11

**Event:** Click the **Start** button.

**Default:** ./.
CHECK_RESERVE

Description: This check procedure is executed during list position processing after the \textbf{Start} button has been clicked. The \textbf{Shortage} run option must be set to \textbf{2-Pass CIP}. The code should be modified according to the customer's business requirements.

Type: Function

Usage: C.30.11

Event: Click the \textbf{Start} button.

Default: 1

COST

Description: This function calculates the cost of an ident. It is a placeholder function with a useful default functionality. The code should be modified according to the customer's business requirements.

Type: Function

Usage: C.30.01, C.30.11

Event: Click the \textbf{Start} button.

Default: \texttt{mvp\_ident\_values.number\_value*1\_quantity} for attribute MATL\_COST

MANHOURS

Description: This function calculates the manhours for an ident. It is a placeholder function with a useful default functionality. The code should be modified according to the customer's business requirements.

Type: Function

Usage: C.30.01, C.30.11

Event: Click the \textbf{Start} button.

Default: \texttt{mvp\_ident\_values.number\_value*1\_quantity} for attribute MANHOURS\_C

80 SmartPlant Materials Customer-Installed Procedures (CIPs)
WEIGHT

Description: This function calculates the weight of an ident. It is a placeholder function with a useful default functionality. The code should be modified according to the customer's business requirements.

Type: Function
Usage: C.30.01, C.30.11
Event: Click the Start button.
Default: m_pck_req_custom.weight(l_ident) * l_quantity

M_PCK_IDENT_CUSTOM

This package contains custom functions and procedures for ident management.

POST_BATCH

Description: This procedure is executed after building idents in batch mode from the S.50.06 Specification Management screen.

Type: Procedure
Usage: S.50.06
Event: Click Build Idents and Batch buttons.
Default: NULL

PRE_BATCH

Description: This procedure is executed before building idents in batch mode from the S.50.06 Specification Management screen.

Type: Procedure
Usage: S.50.06
Event: Click Build Idents and Batch buttons.
Default: NULL
Custom Packages

START_BATCH_BUILD
Description: This procedure is executed when building idents in batch mode from the S.50.06 Specification Management screen.
Type: Procedure
Usage: S.50.06
Event: Click Build Idents and Batch buttons.
Default: m_pck_job_scheduler.create_n_run_job

M_PCK_INTOOLS_CUSTOM
This package contains custom procedures for converting the SmartPlant Instrumentation BOM data into OMI format.

ADD_ATTRS
Description: This CIP procedure is executed after the BOM load has created positions.
Type: Procedure
Usage: B.40.01
Event: Click the Start button.
Default: NULL

PRE_LOADING
Description: This CIP is executed before the BOM load starts.
Type: Procedure
Usage: B.40.01
Event: Click the Start button.
Default: NULL
**M_PCK_INVOICE_CUSTOM**

This package contains custom procedures for invoicing in the MSCM module of SmartPlant Materials.

**GET_IVC_NUMBER**

**Description:** This CIP function is used to get the invoice number for a new invoice.

**Type:** Function

**Usage:** P.50.92

**Event:** Executed when P.50.92 Invoices is opened; defines the initial value of the invoice number.

**Default:** The next available invoice number is generated based on the rule defined by project default ZP_IVCRULE and the range defined by project default ZP_IVCRANG.

**GET_REMIT_ADD**

**Description:** This CIP function is used to get the ‘Remit To’ address.

**Parameters:**

- p_ca_id - ID of the company address that is stored in the table M_INVOICES.

**Type:** Function

**Usage:** P.50.92

**Event:** Executed when the LOV is used in the Remit To field.

**Default:** For the company assigned to the agreement, the company addresses that are marked for payment are returned as the possible values (Address 1, Address 2, City, State, Country, Zip code from m_company_addrs).
SUBMIT_INVOICE

Description: This CIP function is called when an invoice is submitted.

Parameters:

p_ivc_id - ID of the invoice that is submitted (from table M_INVOICES).

Type: Function

Usage: SM.20.11, P.50.92

Event: Click the Submit button.

Default: NULL

UNSUBMIT_INVOICE

Description: This CIP function is called when an invoice is unsubmitted.

Parameters:

p_ivc_id - ID of the invoice that is submitted (from table M_INVOICES).

Type: Function

Usage: SM.20.11, P.50.92

Event: Click the Unsubmit button.

Default: NULL
UPD_INVOICE_STATUS

Description: This CIP procedure is executed when the invoice status is updated.

Parameters:

p_ivc_id - ID of the invoice.
p_status - Value of the invoice status that will be used to set the status in the database.

Possible values for the invoice status are as shown below.

<table>
<thead>
<tr>
<th>Invoice Status in database</th>
<th>Invoice Status on screen</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Not Submitted</td>
</tr>
<tr>
<td>S</td>
<td>Submitted</td>
</tr>
<tr>
<td>NA</td>
<td>Needs Assignment</td>
</tr>
<tr>
<td>P</td>
<td>Pending</td>
</tr>
<tr>
<td>RFA</td>
<td>Ready for Approval</td>
</tr>
<tr>
<td>RFFA</td>
<td>Ready for Final Approval</td>
</tr>
<tr>
<td>A</td>
<td>Approved</td>
</tr>
<tr>
<td>R</td>
<td>Rejected</td>
</tr>
<tr>
<td>PA</td>
<td>Paid</td>
</tr>
</tbody>
</table>

Type: Procedure

Usage: P.50.91, P.50.92, P.50.93

Event: Invoice status update

Default: Status
Custom Packages

M_PCK_ISKETCH_CUSTOM
This package supports functionality around export of spec data to I-Sketch catalog files.

CONVERT_BORE_UNIT_TO_UNIT

Description: This CIP function is called when the export unit is set for an export job on the I-Sketch 20.05 Export Jobs screen. The purpose is to convert all bore values from a source unit to a target unit. The used unit codes for conversion may vary depending on the customer's database. Here unit codes IN and MM are used to identify the correct S.40.04.xx conversion table.

Parameters passed in:

  p_from_unit_id: Current unit ID to be converted
  p_to_unit_id: Unit ID for output
  p_value: Value to be converted

This code is not maintained by Intergraph; as with all CIP code, it is the customer's responsibility to use this API and maintain their custom code. Customers must save their code as text files, because SmartPlant Reference Data and SmartPlant Materials DB migration overwrites it without any warning.

IMPORTANT: This procedure must not commit. Otherwise, the export may not work properly.

Type: Function

Usage: ISK.20.05

Event: During export, for each bore value.

Default: Input value
CONVERT_DIM_UNIT_TO_UNIT

Description: This CIP function is called when the export unit is set for an export job on the I-Sketch 20.05 Export Jobs screen. The purpose is to convert all dimensional values from a source unit to a target unit. The used unit codes for conversion may vary depending on the customer’s database. Here unit codes IN and MM are used to identify the correct S.40.04.xx conversion table.

Parameters passed in:

- p_from_unit_id: Current unit ID to be converted
- p_to_unit_id: Unit ID for output
- p_value: Value to be converted

This code is not maintained by Intergraph; as with all CIP code, it is the customer’s responsibility to use this API and maintain their custom code. Customers must save their code as text files, because SmartPlant Reference Data and SmartPlant Materials DB migration overwrites it without any warning.

**IMPORTANT** This procedure must not commit. Otherwise, the export may not work properly.

Type: Function

Usage: ISK.20.05

Event: During export, for each dimensional value

Default: Input value
CUSTOM PACKAGES

GENERAL_CIP

Description: This CIP is called after collecting data for an export job but prior to generation of export files.

The parameter passed in is:

p_isej_id - Primary key of table m_isk_exp_jobs; identifies the current job uniquely.

For a spec driven export:

- isej_id - Allows you to select specs attached to the job (pk_id = ises_id).
- ises_id - Allows you to select on the commodity level from m_isk_exp_spec_ccs

For a master file export:

- isej_id - Allows you to select from m_isk_exp_master_ccs.

This code is not maintained by Intergraph; as with all CIP code, it is the customer's responsibility to use this API and maintain their custom code.

Customers must save their code as text files, because SmartPlant Reference Data and SmartPlant Materials DB migration overwrites it without any warning.

**IMPORTANT** This procedure must not commit. Otherwise, the export may not work properly.

Type: Procedure

Usage: ISK.20.05

Event: Called once after the export data has been collected but not written to files.

Default: None
GET_CC_DESCRIPTION

Description: This CIP has been incorporated into this package in order to allow customers to fill a user-definable export value for a commodity code description. The returned value is truncated to 512 characters. This CIP is executed by m_pck_isketch during the export for each commodity code.

The parameter passed in is:

p_cc_id  commodity_id from table m_commodity_codes

This code is not maintained by Intergraph; as with all CIP code, it is the customer's responsibility to use this API and maintain their custom code. Customers must save their code as text files, because SmartPlant Reference Data and SmartPlant Materials DB migration overwrites it without any warning.

IMPORTANT: This procedure must not commit. Otherwise, the export may not work properly.

Type: Function

Usage: ISK.20.05

Event: During export, for each commodity code

Default: m_commodity_code_nls.TRIM(SUBSTR(short_desc,1,512))

GET_IDENT_BOLT_DIMS

Description: This CIP is called to derive attribute values for bolt diameter, bolt length, and bolt quantity (BDIA, BLGT, BQTY). Parameters passed in are:

p_ident: Current ident to export
p_interface_id: Stored for interface I-Sketch in table m_interfaces
p_translation: Fixed attribute translations (BDIA, BLGT, BQTY) passed in

This code is not maintained by Intergraph; as with all CIP code, it is the customer's responsibility to use this API and maintain their custom code. Customers must save their code as text files, because SmartPlant Reference Data and SmartPlant Materials DB migration overwrites it without any warning.

IMPORTANT: This procedure must not commit. Otherwise, the export may not work properly.

Type: Procedure

Usage: ISK.20.05

Event: During export, for each ident with SKey 'BOLT'
**Custom Packages**

**Default:** Attribute value read from geometric tables with or without unit conversion

**GET_IDENT_DESCRIPTION**

**Description:** This CIP has been incorporated into this package in order to allow customers to fill a user-definable export value for an ident description. The returned value is truncated to 512 characters. This CIP is executed by m_pck_isketch during the export for each ident.

Parameters passed in are:

- **p_ident**  ident from table m_idents

This code is not maintained by Intergraph; as with all CIP code, it is the customer’s responsibility to use this API and maintain their custom code. Customers must save their code as text files, because SmartPlant Reference Data and SmartPlant Materials DB migration overwrites it without any warning.

**IMPORTANT** This procedure must not commit. Otherwise, the export may not work properly.

**Type:** Function

**Usage:** ISK.20.05

**Event:** During export, for each ident

**Default:** NLS dependent result of SUBSTR( m_commodity_code_nls.short_desc||
  m_ident_nls.short_desc),1,512)

**M_PCK_LOGISTIC_CUSTOM**

This package contains all procedures and functions that can be customized for the logistics screens.
GET_IS_POS

Description: Example of a function for item shipments.
Type: Function
Usage: P.10.74, P.70.61, P.70.62
Event: This example CIP can be integrated in the configuration that is set up on P.10.74 for P.70.62. When you execute a query on P.70.62, the software calls this CIP.
Default: m_item_ships.item_ship_pos

GET_ORDER_NUMBER

Description: Example of a function for orders.
Type: Function
Usage: P.10.74, P.70.61, P.70.62
Event: This example CIP can be integrated in the configuration that is set up on P.10.74 for P.70.62. When you execute a query on P.70.62, the software calls this CIP.
Default: m_po_headers.po_number

GET_SG_CODE

Description: Example of a function for orders with shipment groups.
Type: Function
Usage: P.10.74, P.70.61, P.70.62
Event: This example CIP can be integrated in the configuration that is set up on P.10.74 for P.70.62. When you execute a query on P.70.62, the software calls this CIP.
Default: m_shipment_groups.sg_code
GET_SHIPPING_STATUS

Description: This function determines the shipping status of an item shipment. The function returns:

0 - if the material is already shipped, that is, ATD (actual time of departure) lies in the past,
1 - if the material has not been shipped, that is, no carrier has been assigned,
2 - if the material is next to be shipped, that is, the ETD (estimated time of departure) lies in the future,
9 - in all other cases.

Type: Function
Usage: P.70.62
Event: This sample CIP can be integrated in the configuration that is set up on P.10.74 for P.70.62. When you execute a query on P.70.62, the software calls this CIP.

Default: 1, if m_item_ships.csh_id is NULL (carrier not assigned, material not shipped)
0, if m_item_ships.csh_id is not NULL and m_cargo_ships.atd_date<sysdate
2, if m_item_ships.csh_id is not NULL and m_cargo_ships.etd_date>sysdate
Else 9

P7061_SAMPLE

Description: This procedure is an example for the CIPs that can be started from P.70.61 Logistics Overview.
The name of all these CIPs must start with P7061_

Input parameters are:
- p_poh_id primary key from M_PO_HEADERS
- p_sg_id primary key from M_SHIPMENT_GROUPS

Type: Procedure
Usage: P.70.61
Event: Click the Execute CIP button on P.70.61.

Default: NULL
P7062_SAMPLE

Description: This procedure is an example for the CIPs that can be started from \texttt{P.70.62}

Logistics Details.

The name of all these CIPs must start with \texttt{P7062_}

Input parameters are:
- \texttt{p_item_ship_id} primary key from \texttt{M_ITEM_SHIPS}

Type: Procedure

Usage: \texttt{P.70.62}

Event: Click the \textbf{Execute CIP} button on \texttt{P.70.62}.

Default: NULL

M_PCK_MAIL_CUSTOM

This package contains procedures and functions to generate all e-mail text.

CREATE_MSG_TEXT_INQ

Description: This internal function constructs the text for notification e-mail messages related to the approval/rejection of inquiries.

Type: Function

Usage: \texttt{A.20.49}

Event: Select the \textbf{Approved} or \textbf{Rejected} check box.

Default: See source code.
CREATE_MSG_TEXT_SUP

Description: This internal function constructs the text for notification e-mail messages related to the approval/rejection of bidder lists for inquiries.

Type: Function

Usage: A.20.49

Event: Select the Approved or Rejected check box.

Default: See CIP source code

CREATE_MSG_TEXT_ADD

Description: This internal function constructs the text for notification e-mail messages related to the approval/rejection of addenda.

Type: Function

Usage: A.20.49

Event: Select the Approved or Rejected check box.

Default: See CIP source code

CREATE_MSG_TEXT_PO

Description: This internal function constructs the text for notification e-mail messages related to the approval/rejection of purchase orders.

Type: Function

Usage: A.20.49

Event: Select the Approved or Rejected check box.

Default: See CIP source code
CREATE_MSG_TEXT_QS
Description: This internal function constructs the text for notification e-mail messages related to the approval/rejection of quote summaries.
Type: Function
Usage: A.20.49
Event: Select the Approved or Rejected check box.
Default: See CIP source code

CREATE_MSG_TEXT_REQ
Description: This internal function constructs the text for notification e-mail messages related to the approval/rejection of requisitions.
Type: Function
Usage: A.20.49
Event: Select the Approved or Rejected check box.
Default: See CIP source code

CREATE_MSG_TEXT_Q
Description: This internal function constructs the text for notification e-mail messages related to the approval/rejection of qualifications.
Type: Function
Usage: A.20.49
Event: Select the Approved or Rejected check box.
Default: See CIP source code
CREATE_MSG_TEXT_PNO

Description: This internal function constructs the text for notification e-mail messages related to the approval/rejection of progress numbers.

Type: Function

Usage: A.20.49

Event: Select the Approved or Rejected check box.

Default: See CIP source code

CREATE_MSG_TEXT_SOV

Description: This internal function constructs the text for notification e-mail messages related to the approval/rejection of Schedule of Values (SOV).

Type: Function

Usage: A.20.49

Event: Select the Approved or Rejected check box.

Default: See CIP source code

CREATE_MSG_TEXT_COR

Description: This internal function constructs the text for notification e-mail messages related to the approval/rejection of change order requests (CORs).

Type: Function

Usage: A.20.49

Event: Select the Approved or Rejected check box.

Default: See CIP source code
**CREATE_MSG_TEXT_BC**

Description: This internal function constructs the text for notification e-mail messages related to the approval/rejection of backcharges.

Type: Function

Usage: A.20.49

Event: Select the **Approved** or **Rejected** check box.

Default: See CIP source code

---

**CREATE_MSG_TEXT_NCR**

Description: This internal function constructs the text for notification e-mail messages related to the approval/rejection of non conformance reports (NCR).

Type: Function

Usage: A.20.49

Event: Select the **Approved** or **Rejected** check box.

Default: See CIP source code

---

**CREATE_MSG_TEXT_CLOI**

Description: This internal function constructs the text for notification e-mail messages related to the approval/rejection of closeout item lists (CLOIs).

Type: Function

Usage: A.20.49

Event: Select the **Approved** or **Rejected** check box.

Default: See CIP source code
CREATE_MSG_TEXT_INV

Description: This internal function constructs the text for notification e-mail messages related to the approval/rejection of invoices.
Type: Function
Usage: A.20.49
Event: Select the Approved or Rejected check box.
Default: See CIP source code

CREATE_MSG_TEXT_LR

Description: This internal function constructs the text for notification e-mail messages related to the approval/rejection of login requests.
Type: Function
Usage: A.20.49
Event: Select the Approved or Rejected check box.
Default: See CIP source code

CREATE_MSG_TEXT_CP

Description: This internal function constructs the text for notification e-mails related to the approval/rejection of company profiles.
Type: Function
Usage: A.20.49
Event: Select the Approved or Rejected check box.
Default: See CIP source code
CREATE_MSG_TEXT_MIR

Description: This internal function constructs the text for notification e-mails related to the approval/rejection of material issue reports.

Type: Function

Usage: A.20.49

Event: Select the Approved or Rejected check box.

Default: See CIP source code

GET_REVERSE_BODY_TEXT

Description: This function returns the e-mail body text when an approval gets reversed.

Type: Function

Usage: A.20.49

Event: Click the Reverse Approval button.

Default: 'Approval of '||pk_type ||' '||pk_code ||' has been reversed.'
'Reason:'
p_reason

GET_REVERSE_SUBJECT_TEXT

Description: This function returns the e-mail subject text when approval gets reversed.

Type: Function

Usage: A.20.49

Event: Click the Reverse Approval button.

Default: 'Reverse approval of '||pk_type||' '||pk_code
SEND_NOTIFICATIONS

Description: This procedure sends e-mail notifications related to a single approval step within a multi-step approval process. In a multi-step approval process, the users who must approve (or reject) the object in question are ordered by an ordering sequence and may have a distribution category associated with them.

This procedure is called whenever one of those users issues or withdraws an approval or a rejection and sends a notification e-mail message:

a) to the next user who should (not necessarily must) become active within the ongoing approval process, as indicated by his/her order sequence number, but only upon an approval or rejection (that is, a reminder e-mail);

b) to each of the members of the distribution category associated with the user who has issued his/her approval or rejection or the withdrawal of either (for information purposes).

Type: Procedure
Usage: A.20.49, P.30.21, SM.20.11
Event: Select the Approved or Rejected check box.
Default: See CIP source code

M_PCK_MENU_DICT_CUSTOM

This package contains dictionary CIPs.

POPULATE_AUTOCORRECT_TAB

Description: This procedure contains the dictionary for Autocorrect words.

Type: Procedure
Usage: Search for box in the menu tree
Event: Click the Search button.
Default: See CIP source code
POPULATE_MISSPELL_TAB

Description: This procedure contains the dictionary for misspelled words.
Type: Procedure
Usage: Search for box in the menu tree
Event: Click the Search button.
Default: See CIP source code

M_PCK_MS_CUSTOM

Using this Milestones Custom Package, you can implement all CIP PL/SQL functions for a functional date derivation rule definition.

CALC_ACTUAL_DATE_MASTER

Description: This function is called whenever the actual date of a detail milestone has changed. It recalculates and returns the actual date of the corresponding master milestone (specified by parameter 'p_ms_id') based on the change applied to its detail milestone(s).
Type: Function
Usage: P.60.01, P.60.03, P.60.05
Event: Click Calculate Forecasted or Calc All Forecasted on the above mentioned screens, particularly on every database commit that updates a date column of a 'trackable' table in the database.
Default: MIN(m_ms_details.actual_date) WHERE m_ms_details.actual_date IS NOT NULL
CALC_PLANNED_ISSUE_DATE

Description: This procedure calculates the planned issue date for supplement milestone plans of planned requisitions. Depending on the implementation, the value returned by the calculation can be interpreted as days or weeks (default).

Type: Function

Usage: P.60.03

Event: Performed for each record in the Planned Supplements block.

Default: m_pck_calendar.calc_work_day(TRUNC(p_ros_date),(p_lead_time*7))

GET_MS_VAL

Description: This function sets the default milestone set value based on the tagged item type, which can be used while planning entries are automatically being created on P.60.01.

Type: Function

Usage: P.60.01

Event:

Default: m_ms_sets.mset_id

SAMPLE_FNC

Description: Example function providing the signature required for all functions that are to be used for function-based date derivation.

Type: Function

Usage: P.60.01, P.60.03, P.60.05

Event: Click Calculate Forecasted or Calc All Forecasted on the above mentioned screens, particularly on every database commit that updates a date column of a 'trackable' table in the database.

Default: SYSDATE
Custom Packages

SET_MS_PLANNING_DATE

Description: This procedure sets a planning date for the milestones. By default, the procedure sets ‘SYSDATE’ as the planning date for the initial milestone.

Type: Procedure

Usage:

Event:

Default: UPDATE m_milestones SET scheduled_date = SYSDATE

M_PCK_MTI_CUSTOM

This package supports custom code for publishing to SmartPlant.

FILL_PO_UDF

Description: This CIP procedure has been incorporated into this package in order to allow customers to fill the user-definable fields UDF1 to UDF5 VARCHAR2(4000) for each purchase order publish record. Each of the user-definable fields can contain up to 4000 characters. This procedure is executed by m_pck_mti_publish when a purchase order is approved.

Parameters passed in are:

- p_poh_id  Purchase Order header ID from table m_po_headers
- p_spop_id  Updatable record of publish table m_spf_po_publishes

This code is not maintained by Intergraph; as with all CIP code, it is the customer’s responsibility to use this API and maintain their custom code. Customers must save their code as text files, because SmartPlant Reference Data and SmartPlant Materials DB migration overwrites it without any warning.

IMPORTANT: This procedure must not commit. Otherwise, the export may not work properly.

Type: Procedure

Usage: P.50.07

Event: Approve a purchase order.

Default: UPDATE m_spf_po_publishes SET UDF1-5 = './.'
**FILL_REQ_UDF**

**Description:** This CIP procedure has been incorporated into this package in order to allow customers to fill the user-definable fields UDF1 to UDF5 VARCHAR2(4000) for each purchase order publish record. Each of the user-definable fields can contain up to 4000 characters. This procedure is executed by m_pck_mti_publish when a purchase order is approved.

Parameters passed in are:
- `p_r_id` Requisition ID from table m_reqs
- `p_srqp_id` Updatable record of publish table m_spf_req_publishes

This code is not maintained by Intergraph; as with all CIP code, it is the customer's responsibility to use this API and maintain their custom code. Customers must save their code as text files, because SmartPlant Reference Data and SmartPlant Materials DB migration overwrites it without any warning.

**IMPORTANT:** This procedure must not commit. Otherwise, the export may not work properly.

**Type:** Procedure

**Usage:** R.30.01

**Event:** Release to procurement

**Default:** UPDATE m_spf_req_publishes SET UDF1-5 = './.'

---

**M_PCK_OMI_CUSTOM**

This CIP package fills the OMI interface table.

---

**BOM_NODE_CREATED**

**Description:** Allows any user-definable action after a new node has been created by OMI.

**Type:** Procedure

**Usage:** B.40.01

**Event:** During OMI import, called after a new node has been created

**Default:** NULL
**BOM_NODE_FOUND**

Description: Allows any user-definable action after a node has been found by OMI.

Type: Procedure

Usage: B.40.01

Event: During OMI import, called after a node has been found

Default: NULL

**BOM_NODE_PRE_TSQ**

Description: Called by m_pck_bom_load for each position of the current node that is affected by TSQ (issued_qty or resv_qty > 0 or placed on an unposted MIR).

Type: Procedure

Usage: B.40.01

Event: For each position before TSQ is called.

Default: NULL

**BOM_POS_CREATED**

Description: Allows any user-definable action after a new position has been created by OMI.

Type: Procedure

Usage: B.40.01

Event: During OMI import, called after a position has been created

Default: NULL
CHECK_CONSISTENCY

Description: Pre-OMI CIP, called from omi_job_start.
If the spec code and ident code exist in m_import_interfaces, then check if the ident exists in the spec. If the ident does not exist, then set ident code = NULL.
If the spec code, commodity code, and size1 (size2, schedules) exist in m_impost_interfaces, then check if one ident for this commodity code (with sizes and schedules) exists. If the one ident exists, then set the ident code.

Type: Procedure
Usage: B.40.01
Event: Called during the OMI import job when table M_IMPORT_INTERFACES has been filled but processing has not started yet.
Default: See CIP source code

OMI_EDIT_FLAT_PRE_IMPORTS

Description: This procedure is intended to perform any changes concerning table m_flat_pre_imports.

Type: Procedure
Usage: B.40.01
Event: Called during OMI by FLC when table M_FLAT_PRE_IMPORTS is filled but not yet processed and written to M_IMPORT_INTERFACES.
Default: NULL

OMI_FIND_CC_BY_TD

Description: This procedure finds and sets a commodity code for records with tables and table details specified. Only records without an ident and commodity code will be updated.

Type: Procedure
Usage: B.40.01
Event: Called for OMI by FLC, once at the end of filling table M_FLAT_PRE_IMPORTS.
Default: See CIP source code

**OMI_FLC_LOAD**

Description: This procedure allows you to fill in FLC data by using your own PL/SQL code.
Type: Procedure
Usage: B.40.01
Event: Click the **Start** button.
Default: NULL

**OMI_JOB_AFTER_COPY**

Description: This procedure is executed by the **New Run** button on B.40.01.
Type: Procedure
Usage: B.40.01
Event: Click the **New Run** button.
Default: NULL

**OMI_JOB_END**

Description: This procedure is called after OMI completes import processing but before clearing table M_IMPORT_INTERFACES. So you can still access this data and also data that has been produced by the import.
Type: Procedure
Usage: B.40.01
Event: Called at the end of OMI data import.
Default: NULL
Custom Packages

**OMI_JOB_START**

Description: This procedure is called after table M_IMPORT_INTERFACES is filled but before OMI starts to process this data.

Type: Procedure

Usage: B.40.01

Event: Called at the beginning of OMI data processing.

Default: NULL

**OMI_LOAD**

Description: This procedure populates the interface table for OMI jobs with ‘by OMI CIP’ in the Data Source field.

Type: Procedure

Usage: B.40.01

Event: Called by clicking the Start button for this particular import type.

Default: NULL (example code available)

**OMI_POST_CHANGES_AFTER_IMPORT**

Description: This procedure performs any necessary changes after the FLC BOM import.

Type: Procedure

Usage: B.40.01

Event: Called at the end of an FLC based OMI import.

Default: NULL (example code available)
**OMI_SET_ITEM_TYPE**

Description: This procedure sets the item type accordingly. This procedure affects only records without an item type filled by the OMI job.
- Ident and tag number filled --> item rule SWT applies.
- Only tag number filled --> item rule TWM applies.
- Only Ident filled --> ignore because default will be set by OMI.

Type: Procedure

Usage: B.40.01

Event: Called during OMI when table M_IMPORT_INTERFACES is already filled but processing has not yet started.

Default: See CIP source code

---

**OMI_SET_UNIT**

Description: This procedure sets the units in m_flat_pre_imports if not assigned. The values of the parameters p_qty_unit and p_len_unit are defined as "Quantity Unit" and "Length Unit" on the B.10.41 Flat Configurations screen.

Type: Procedure

Usage: B.40.01

Event: Called during FLC-based OMI to update values on table m_flat_pre_imports.

Default: NULL (example code available)

---

**M_PCK_OPI_CUSTOM**

This package contains customizable procedures and functions for OPI (Open Procurement Interface).
ASCII_EXAMPLE

Description: Example procedure to load an ASCII file. This procedure reads the data from an ASCII file and stores its elements in the M_OPI_IMPORT_DATA table.

Type: Procedure

Usage: OPI.10.01

Event: Click the Start button to start the import job.

Default: Example code to read an ASCII file and store data in M_OPI_IMPORT_DATA

DATA_EXAMPLE

Description: Example CIP procedure to load data. This procedure reads the data from M_OPI_IMPORT_STRINGS and stores its elements in the M_OPI_IMPORT_DATA table.

Type: Procedure

Usage: OPI.10.01

Event: Click the Start button to start the import job.

Default: Example code to read M_OPI_IMPORT_STRINGS and store data in M_OPI_IMPORT_DATA

POST_OPI_IMPORT

Description: This CIP is executed after the OPI import has been performed.

Type: Procedure

Usage: OPI.10.01

Event: Click the Start button to start the import job.

Default: NULL
XML_EXAMPLE

Description: Example CIP procedure to load an XML file. This procedure reads the data from an XML file and stores its elements in the M_OPI_IMPORT_DATA table if parsing was successful.

Type: Procedure

Usage: OPI.10.01

Event: Click the Start button to start the import job.

Default: Example code to read an XML file and store data in M_OPI_IMPORT_DATA

M_PCK_OSCI_CUSTOM

This package contains customizable procedures and functions for OSCI (Open Supplier/Company Interface).

XML_EXAMPLE

Description: Example CIP procedure to load an XML file. This procedure reads the data from an XML file and stores its elements in the M_OSCI_IMPORT_DATA table if parsing was successful.

Type: Procedure

Usage: OSCI.10.01

Event: OSCI data import

Default: ./ (see source code)
M_PCK_PDMSEXP_CUSTOM

This package is a placeholder for CIP procedures related to PDMS export.

PUBLISH

Description: This procedure is executed after a PDMS job has finished successfully.
Type: Procedure
Usage: PDMS.10.02
Event: Click the **Transfer selected Specs** button.
Default: `M_PCK_AUDIT_USERS.insert_audit_trail('M_PDMS_LOGS',TO_CHAR(p_id), 'I','PDMS Export CIP called',SYSDATE)`

M_PCK_PDMS_CUSTOM

This package contains customizable procedures and functions for the PDMS interface.

BFLE_WEIGHT

Description: This function retrieves the bfile weight; it is customized for SDB Specon.
Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: `p_weight`
BOLT_HANDLING

Description: This procedure is used for handling bolts to delete multiple occurrences of the same BDIA (bolt diameter). These occurrences are checked within a spec, GType config, and commodity code combination. A bolt is recognized if GType="BOLT".

Type: Procedure
Usage: PDMS.20.01
Event: Create PDMS files
Default: See CIP source code

CATE

Description: This function returns the interface commodity code.

Type: Function
Usage: PDMS.10.02, PDMS.20.01
Event: Create PDMS files
Default: SUBSTR(m_api_CC.get_cci_code_for(p_commodity_id,p_interface_id),1,20)

CATE_STEXT

Description: This function returns the geometric description.

Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: mvp_geom_name_nls.description
Custom Packages

**CATE_TEXT**

Description: This function returns the geometric name.
Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: mvp_geom_names.gn_code

**CHECK_EXPORT_ALLOWED**

Description: This function checks whether a specification can be exported.
Type: Function
Usage: PDMS.10.02
Event: Create PDMS specification files
Default: Y when spec is found; N when spec is not found.

**CHECK_IDENTS**

Description: This function checks the idents for validity.
Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: 1
**CLEAN_SORT_ALONG_LINE**

Description:

Type: Procedure

Usage: PDMS.20.01

Event: Create PDMS files

Default: DELETE m_pdms_sort_spec_exports WHERE ((log_id=p_log_id) Or (LMOD < SYSDATE -20)) AND ROWNUM <1000;

---

**COCO**

Description:

Type: Function

Usage: PDMS.20.01

Event: Create PDMS files

Default:

---

**DTSE**

Description:

Type: Function

Usage: PDMS.20.01

Event: Create PDMS files

Default:
Custom Packages

example_cip_pdms3002

Description: Provides an example for a CIP function that is used on PDMS 30.02 G-Type Attribute Mapping. The input parameter must be named p_ident. The return value is written to the PDMS_G.DAT file as VARCHAR2(19).

Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: 'CIP example'

FILE_BOLT_REF

Description: This procedure identifies the bolt reference data.

Type: Procedure
Usage: PDMS.20.01
Event: Create PDMS files
Default: See CIP source code

FILL_COMP_ATTRS

Description:

Type: Procedure
Usage: PDMS.20.01
Event: Create PDMS files
Default:
**FIND_IN_CCD**

Description: This function fills M, X, Y and ZTEXT in PDMS_M.dat.

Type: Function

Usage: PDMS.20.01

Event: Create PDMS files

Default: See CIP source code

**GENERAL_PDMS2001**

Description: General procedure customized for SDB PDMS_M.dat, PDMS_BS.dat, PDMS_B.dat.

Type: Procedure

Usage: PDMS.20.01

Event: Create PDMS files

Default: See CIP source code

**GENERAL_SPEC_FILE_UPDATE**

Description:

Type: Procedure

Usage: PDMS.20.01

Event: Create PDMS files

Default: DELETE mvp_pdms_link_scoms_specs

Insert into mvp_pdms_link_scoms_specs …

UPDATE MV_PDMS_SPEC_identS set CATREF = ‘=0’

UPDATE MV_PDMS_SPEC_identS set CMPPREF = ‘=0’
GET_ATTR_VALUE

Description: This function returns the value of an attribute mapped to the physical attribute (of table m_pdms_spec_ids) on the PDMS 10.06 screen. If it is not a standard attribute, that is, if it is a user-defined attribute with an attribute code other than 'ANGLE', 'BLTREF', 'CATREF', 'CMPPREF', 'DETAIL', 'PBOR0', 'PBOR1', 'PBOR2', 'PBOR3', 'PBOR4', 'RADIUS', 'STYP', 'SUFX', 'BDIA', 'CONX', 'BTYP', 'BAUL', 'RATI', 'SHOP', 'MATXT', 'UNIT0', 'UNIT1', 'UNIT2', 'UNIT3', 'UNIT4', or 'UNITBDIA', then the value is set by this function.

Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: '!NDF'

GET_BTYPE

Description: This function returns the BTYPE.

Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: 'BTYPE BOLT' if m_pck_pdms.ident_attr_value(p_ident, p_attr_id) is NULL else 'BTYPE'|| m_pck_pdms.ident_attr_value(p_ident, p_attr_id)

GET_CATA

Description: SDB does not use CATA, so this procedure returns SDB_STD_DESC.

Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: See CIP source code
GET_FLBP

Description:

Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: 0

GET_IDENT_CODE

Description: This function returns the ident code for a given ident.

Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: mvp_idents.ident_code

GET_INSU_CATREF

Description: This function returns the necessary catref reference for the Specon file.

Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: '/IS' || p_insu_thk_value
Custom Packages

GET_INSU_DETAIL
Description: This function returns the necessary detail reference for the Specon file.
Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: '/DE.INS'

GET_INSU_G_DAT
Description: This function returns the necessary values for the insulation _g.dat file.
Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: see source code

GET_INSU_HEADING
Description: This function returns the necessary heading fields for the Specon file.
Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: 'TYPE NAME TEMP PBOR0 CATREF DETAIL'
GET_INSU_NAME
Description: This function returns the necessary name reference for the Specon file.
Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: ‘*/IN’ || p_gd_id

GET_INSU_PBOR0
Description: This function returns the necessary pbor0 for the Specon file.
Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: p_dn_FROM_value || p_unit || ',' || p_dn_to_value || p_unit

GET_INSU_TEMP
Description: This function returns the necessary temp for the Specon file.
Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: p_temp_FROM_value || ',' || p_temp_to_value
GET_INSU_TYPE
Description: This function returns the necessary g-type reference for the Specon file.
Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: ‘INSU’

GET_PDMS_INTERFACE_ID
Description: This function returns the ID for the PDMS interface defined on A.70.01.
Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: m_interfaces.interface_id where interface_code='PDMS'

GET_UNIT
Description: This function returns the unit for a given commodity code.
Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: NULL
GPD_REF

Description:
Type: Procedure
Usage: PDMS.20.01
Event: Create PDMS files
Default:
\[
\begin{align*}
l\text{_cate} & := m\_pck\_pdms\_custom\.cate(p\_interface\_id, p\_commodity\_id); \\
p\_gmref & := l\_cate; \\
p\_tref & := l\_cate; \\
p\_dref & := l\_cate
\end{align*}
\]

GTYPE

Description: This function returns the GTYPE.
Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: \( m\_pck\_pdms\_custom\.s2008(p\_commodity\_id,'GTYPE', p\_interface\_id) \)

IDENT_ATTRS

Description:
Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: See CIP source code
**PREP_OUT_SET_DELETE_DESC**

Description: This function returns a header line for the delete file. The function is called in `m_pck_pdms_spec.prepare_output`.

Type: Function

Usage: PDMS.20.01

Event: Create PDMS files

Default: "DESC "Deleted Date:"||v_date||"

**PREP_OUTPUT**

Description: This procedure is used to prepare the output file. Customized for SDB Specon.

Type: Procedure

Usage: PDMS.20.01

Event: Create PDMS files

Default: Sample code available.

**PREP_OUTPUT_DEL**

Description: This procedure is used to prepare the delete file. Customized for SDB Specon.

When a delta file of delete is created, the extra data also appears.

Type: Procedure

Usage: PDMS.20.01

Event: Create PDMS files

Default: NULL
PREP_OUTPUT_HEADING

Description: This procedure is used to prepare the heading of the output file. Customized for SDB Specon.

Type: Procedure
Usage: PDMS.20.01
Event: Create PDMS files
Default: See CIP source code

PREP_OUTPUT_INS

Description: This procedure is used to prepare the insert file. Customized for SDB Specon. When a delta file of insert is created, the extra data also appears.

Type: Procedure
Usage: PDMS.20.01
Event: Create PDMS files
Default: m_pck_m.ml('DEBUG: PREP_OUTPUT_INS P_LOGIN:'||p_log_id,0,FALSE)

PREP_OUTPUT_MOD

Description: This procedure is used to prepare the modify file. Customized for SDB Specon. When a delta file of modify is created, the extra data also appears.

Type: Procedure
Usage: PDMS.20.01
Event: Create PDMS files
Default: m_pck_m.ml('DEBUG: PREP_OUTPUT_MOD P_LOGIN:'||p_log_id,0,FALSE)
Custom Packages

S2008
Description: Based on the mapping defined on S.20.08, this function returns the mapped value for a commodity code.
Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: mvp_table_details.td_code

SCOM
Description: 
Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: See CIP source code

SECT
Description: Customized for SDB detail.
Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: 'SDTEXT'
SET_INSU_FIELDS
Description: This function returns the necessary heading fields for the Specon file.
Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: p_type || ' ' || p_name || ' ' || p_temp || ' ' || p_pbor0 || ' ' || p_catref || ' ' || p_detail;

SET_S2008
Description: This CIP generates Specon files. Customized for SDB Specon.
Type: Procedure
Usage: PDMS.20.01
Event: Create PDMS files
Default: See CIP source code

SET_SIZE_CONVERSION
Description:
Type: Function
Usage: PDMS.20.01
Event: Create PDMS files
Default: m_nominal_size_details.UPPER(SUBSTR(short_desc,1,1))
**SKEY**

Description: PDMS requires SKEY to generate the ISO symbol. Normally, SKEY is defined on S.20.08, dependent on group, part, and end prep.

Type: Function

Usage: PDMS.20.01

Event: Create PDMS files

Default: See CIP source code

**SORT_ALONG_LINE**

Description: This procedure creates sorting along the line content.

Type: Procedure

Usage: PDMS.20.01

Event: Create PDMS files

Default: See CIP source code

**SPEC_BOLT_REF**

Description:

Type: Procedure

Usage: PDMS.20.01

Event: Create PDMS files

Default: NULL
TEXT

Description: Detail text (Ident Description) should be the same as displayed on the SmartPlant Materials screens. The function m_pck_std_custom.ident_desc is used. Only bolt is used for the commodity code description (because PDMS puts the bolt dia and length in ISO).

Type: Function

Usage: PDMS.20.01

Event: Create PDMS files

Default: See CIP source code

M_PCK_PDSEXP_CUSTOM

This package contains CIPs related to PDS export.

PUBLISH

Description: This procedure is executed after the PDS job has finished successfully.

Type: Procedure

Usage: PDS.10.12

Event: Click the Create PMC/PCD button.

Default: M_PCK_AUDIT_USERS.insert_audit_trail('M_PDS_PMC', TO_CHAR(p_pmc_id), 'I', 'PDS Export called', SYSDATE)
M_PCK_PDS_BOM_CUSTOM

This package provides CIPs called by m_pck_pds_bom_import.

CREATE_OR_SUM_SPECIAL_ITEMS

Description: This function is executed by the standard procedure whenever a position will be created. This function replaced the standard-insert into m_import_interfaces. When the return-value is 1, no insert for this line item from m_pds_raw_data into m_import_interfaces will be made.

Type: Function
Usage: B.40.01
Event: Called during PDS-based OMI whenever a position is created.
Default: 0

GET_LN_CODE

Description: This function is called by m_pck_pds_bom_import, before the preprocessor writes the node into m_import_interfaces.

Type: Function
Usage: B.40.01
Event: During PDS-based OMI on insert into table M_IMPORT_INTERFACES.
Default: The same as the node input value

INSERT_NODE_NLS

Description: This function is called by m_pck_pds_bom_import, right after the preprocessor writes the node into m_import_interfaces.

Type: Procedure
Usage: B.40.01
Event: During PDS-based OMI on insert into table M_IMPORT_INTERFACES.
Default: Node name
**M_PCK_PDS_CUSTOM**

This package contains CIPs for the PDS interface.

**AFTER_PDL_GENERATION**

Description: This procedure runs after a pdl generation run has finished.
Type: Procedure
Usage: PDS.10.12
Event: PDL Generation
Default: See CIP source code

**BOLT_LOGIC**

Description: This procedure handles bolt logic within a specification.
Type: Procedure
Usage: PDS.10.12
Event: Click the Create PMC/PCD button.
Default: See CIP source code

**COMMODITY_DESCRIPTION**

Description: This function returns the description for a commodity code. There are four samples that can be used, or you can create your own.
Type: Function
Usage: PDS.10.10, PDS.10.12
Event: 
Default: m_commodity_code_nls.short_desc
GASKET_LOGIC

Description: This procedure handles gasket logic within a specification.
Type: Procedure
Usage: PDS.10.12
Event: Click the Create PMC/PCD button.
Default: See CIP source code

GET_COMMODITY_CODE

Description: This function is called from cr_pds.sql skript to set m_pds_pcd.free1 and on the PDS 10.12 screen to look up the commodity code for the commodity_id. You can use this function to make final modifications to the column free1, for example, to put/remove flags like asterisks or plus symbols in front of the commodity code, depending on your logic for implied material/assemblies.
The function receives:
- the actual pmc_id
- SmartPlant Materials company_id
- the actual company_id in PDS config
- the transtype where 1 means commodity level and 2 means ident level
- free4, which contains A for assembly, C for an assembly position, and I for a non-assembly item
- the commodity_id from the pcd table
- the ident from the pcd_table, which is only filled when the lines are exploded to the ident level

Type: Function
Usage: PDS.10.12
Event:
Default: See CIP source code
**GET_ENDTYPE**

Description: This function fetches the endprep type for a given endprep. This function is used in m_pck_pds_rules.

Type: Function

Usage: PDS.10.05

Event: 1, 6, 5, or 4 depending on the given endprep

---

**GET_FL_WITH_IO_LINE_IND**

Description: This function controls whether the line is added to the spec-dependent fluid list (1), the project-dependent fluid list generated during spec transfer on PDS.10.12 (2), or the spec-dependent fluid list generated on PDS.10.10 (3).

Type: Function

Usage: PDS.10.10, PDS.10.12

Event: '0'

---

**GET_PDLH_CC**

Description: This function retrieves the commodity code for use in the PDL header name created during generation on PDS 20.10.

Type: Function

Usage: PDS.20.10

Event: PDL creation

Default: m_pck_bom_download.find_cc(p_commodity_id)
**GET_PDLH_ENDPRP**

**Description:** This function retrieves end preparation 1 or 2 for use in the PDL header name created during generation on PDS 20.10.

**Type:** Function

**Usage:** PDS.20.10

**Event:** PDL creation

**Default:** `m_pck_mlcl.get_cgrp(p_commodity_id,p_1_or_2,p_interface_id)`

---

**GET_PDLH_GEOSTD**

**Description:** This function retrieves the geometric standard for use in the PDL header name created during generation on PDS 20.10.

**Type:** Function

**Usage:** PDS.20.10

**Event:** PDL creation

**Default:** `m_pck_mlcl.get_geostd(p_commodity_id, p_cl575_transl, p_cl575_method, p_cl575_start, p_interface_id)`

---

**GET_PDLH_MATERIAL**

**Description:** This function retrieves the material for use in the PDL header name created during generation on PDS 20.10.

**Type:** Function

**Usage:** PDS.20.10

**Event:** PDL creation

**Default:** `m_pck_std.find_in_cc(p_commodity_id, p_model_method, p_cl145_transl)`
GET_PDLH_MODEL

Description: This function retrieves the model code for use in the PDL header name created during generation on PDS 20.10.

Type: Function

Usage: PDS.20.10

Event: PDL creation

Default: m_pck_mlcl.get_model(p_commodity_id, p_model_method, p_part_id, p_interface_id)

GET_PDLH_MODIF

Description: This function retrieves the modifier for use in the PDL header name created during generation on PDS 20.10.

Type: Function

Usage: PDS.20.10

Event: PDL creation

Default: m_pck_mlcl.get_via_cc_or_part(p_commodity_id, p_cl550_transl, p_interface_id)

GET_PDLH_RATING

Description: This function retrieves rating 1 or 2 for use in the PDL header name created during generation on PDS 20.10.

Type: Function

Usage: PDS.20.10

Event: PDL creation

Default: m_pck_mlcl.get_via_cc_or_part(p_commodity_id, p_rat_transl1, p_interface_id)
Custom Packages

GET_PDLH_SCHEDULE
Description: This function retrieves schedule 1 or 2 for use in the PDL header name created during generation on PDS 20.10.
Type: Function
Usage: PDS.20.10
Event: PDL creation
Default: p_input_schedule

GET_PDLH_TS
Description: This function retrieves the table suffix for use in the PDL header name created during generation on PDS 20.10.
Type: Function
Usage: PDS.20.10
Event: PDL creation
Default: m_pck_mlcl.get_ts(p_commodity_id,p_spec_item_id,p_ts_method,p_ts_transl, p_1_or_2, p_cl576_start, p_cl576_start, p_interface_id)

GET_PDLH_WC
Description: This function retrieves the weight code for use in the PDL header name created during generation on PDS 20.10.
Type: Function
Usage: PDS.20.10
Event: PDL creation
Default: m_pck_mlcl.get_wc(p_commodity_id, p_cl578_method, p_wc_source, p_interface_id)
GET_SDD_LINE

Description: This function generates a line of size dependent data (SDD).
Type: Function
Usage: PDS.10.10
Event: Creation of size dependent data (SDD)
Default: See CIP source code

IDENT_DESCRIPTION

Description: This function returns the description for an ident. There are six samples that can be used, or create your own.
Type: Function
Usage: PDS.10.10, PDS.10.12
Event:
Default: m_commodity_code_nls.short_desc||m_ident_nls.short_desc

M_PCK_PID_SPEC_UTL_CUSTOM

This package contains functions for P&ID interface implementation.

GET_COMMODITYOPTION_LONGTXT

Description: This function returns the commodity option long description value for a given code list value from the CommodityOption template sheet, defined in the template that is specified by project default ZI_SP3D_CT.
Type: Function
Usage: This function is used in the P&ID interface to get the value for the commodity option long text from SmartPlant Reference Data.
Custom Packages

Event: This function is called when the get commodity options button of the P&ID interface is clicked. On click of this button, the software calls the getcommodityoptions function, which in turn calls the Get_CommodityOption_LongTxt function.

Default: Attribute value for COMMODITYOPTION LONGDESCRIPTION

GET_COMMODITYOPTION_SHORTTXT

Description: This function returns the commodity option short description value for a given code list value from the CommodityOption template sheet, defined in the template that is specified by project default ZI_SP3D_CT.

Type: Function

Usage: This function is used in the P&ID interface to get the value for the commodity option short text from SmartPlant Reference Data.

Event: This function is called when the get commodity options button of the P&ID interface is clicked. On click of this button, the software calls the getcommodityoptions function, which in turn calls the Get_CommodityOption_ShortTxt function.

Default: Attribute value for COMMODITYOPTION SHORTDESCRIPTION

GET_MAXTEMP

Description: This function takes the specification name with its revision, first size, first size unit, second size, second size unit, short code, first size and second size unit IDs, maximum temperature, and option code as input. This function converts the entered maximum temperature to the same unit as that of the temperature units specified for the specification, and then compares the converted input temperature against the temperatures of the spec and returns one of the following error codes:

1. If any database error occurs while fetching the max temperatures.
2. If the entered maximum temperature is not within the temperature limits of the spec.
3. If the unit specified in SmartPlant Reference Data for the spec is not a valid one.

This function is called from m_pck_pid_spec_utl .getcommoditycode, which is used to get the commodity code.

Type: Function
Usage: This function is used to validate the input maximum temperature from the P&ID interface against the SmartPlant Reference Data maximum temperature specified for a spec.

Event: This function is called when the get commodity code button of the P&ID interface is clicked. On click of this button, the software calls the getcommoditycode function, which in turn calls the Get_MaxTemp function.

Default: 0

---

**GET_MINMAXTEMP**

Description: This function takes specification name with its revision, first size, first size unit, second size, second size unit, short code, first size and second size unit IDs, minimum temperature, maximum temperature, and option code as input. This function converts the entered minimum and maximum temperatures to the same unit as that of the temperature units specified for the specification, and then compares these temperatures against the temperatures of the spec and returns one of the following error codes:

1  => If any database error occurs while fetching min, max temperatures.
2  => If the entered minimum and maximum temperatures are not within the temperature limits of the spec.
5  => If the unit specified in SmartPlant Reference Data for the spec is not a valid one.

This function is called from m_pck_pid_spec_utl.getcommodityoptions, which is used to retrieve the option codes.

Type: Function

Usage: This function is used to validate the input maximum and minimum temperatures from the P&ID interface against the SmartPlant Reference Data maximum and minimum temperatures specified for a spec.

Event: This function is called when the get commodity options button of the P&ID interface is clicked. On click of this button, the software calls the getcommodityoptions function, which in turn calls the Get_MinMaxTemp function.

Default: 0
**GET_SCHEDULE_CDLST**

**Description:** This function returns the codelist number for a given schedule thickness short description value from the ScheduleThickness template sheet, defined in the template that is specified by project default ZI_SP3D_CT.

**Type:** Function

**Usage:** This function is used to get the first size schedule and second size schedule for a spec.

**Event:** This function is called when the get commodity code button of the P&ID interface is clicked. On click of this button, the software calls the getcommoditycode function, which in turn calls the Get_Schedule_CdLst function.

**Default:** 0

---

**GET_SPRD_SHORT_CODE_FORPIPE**

**Description:** This function returns the SmartPlant Reference Data short code for PIPE. This function is called from m_pck_pid_spec_utl.getallowablenominaldiameters where the nominal pipe diameter related to PIPE is determined.

**Type:** Function

**Usage:** This function is used to get the short code for PIPE stored in SmartPlant Reference Data.

**Event:** This function is called when the getnpd button of P&ID interface is clicked. On click of this button, the software calls getallowablenominaldiameters, which in turn calls get_spred_short_code_for_pipe.

**Default:** 0
GET_TEMPLPHYATTR

Description: This function gets the physical attribute that is configured for an attribute in a specification item. This function is called by m_pck_pid_spec_utl.getcommodityoptions.

Type: Function

Usage: This function gets the physical attribute that is configured for an attribute in a spec item of SmartPlant Reference Data.

Event: This function is called when the get commodity options button of the P&ID interface is clicked. On click of this button, the software calls the getcommodityoptions function, which in turn calls the Get_TempPhyAttr function. Because the getcommodityoptions function needs to validate the input maximum temperature, it requires the physical attribute of the temperature configured for the spec in SmartPlant Reference Data, so it calls the Get_TempPhyAttr function.

Default: Physical attribute that is configured for an attribute in a spec item.

PRESSURE_CONVERT_TO_PASCAL

Description: This function returns the conversion factor needed to convert from any unit of pressure to Pascal. The function takes the unit of pressure as input and returns a conversion factor as output. This function is called from the m_pck_pid_spec_utl .validateservicelimits function where it validates whether the entered pressure temperature limits for the spec are within its limits.

Type: Function

Usage: This function is used by the P&ID interface to retrieve the conversion factor to convert the pressure in SmartPlant Reference Data to the standard unit of pressure, that is, Pascal.

Event: This function is called when the validate service limits button of the P&ID interface is clicked. On click of this button, the software calls the validateservicelimits function, which in turn calls the PRESSURE_CONVERT_TO_PASCAL function. Because the validateservicelimits function receives the pressure in Pascal as input from the P&ID interface, but the pressure in SmartPlant Reference Data may not be in Pascal, so to convert it to Pascal, it calls this function so that it can get the conversion factor.

Default: Conversion factor (to Pascal) based on the pressure unit passed.
**TEMP_CONVERT**

**Description:** This function takes a temperature (entered in Kelvin) and unit to which this input temperature is to be converted as input. Based on the unit passed, it converts the input temperature to that unit and returns the converted value. This function returns -1 if the entered unit is not a valid temperature unit. This function is called from m_pck_pid_spec_utl.validateservicelimits.

**Type:** Function

**Usage:** This function is used by the P&ID interface to convert the input temperature entered by the P&ID into the temperature unit present in SmartPlant Reference Data.

**Event:** This function is called when a temperature needs to be converted to another temperature based on the temperature unit passed. This function is called when the validate service limits, get commodity options, or get commodity code buttons are clicked on the P&ID interface.

**Default:** 0

---

**M_PCK_PLATE_CUSTOM**

This package contains procedures for nested system data.

---

**CHECK_DATA**

**Description:** This function checks whether the plate number is already nested. The function returns a numerical value.

**Type:** Function

**Usage:** C.20.15

**Event:** Click the Export button on C.20.15 and the Start button on Window 4.

**Default:** ./.
CREATE_FILE
Description: This procedure writes data for the nested system to the export file.
Type: Procedure
Usage: C.20.15
Event: Click the Export button on C.20.15 and the Start button on Window 4.
Default: ./.

M_PCK_PLL_VERIFY_CUSTOM
This package contains all procedures and functions for the PLL (process line list). The CIPs must be activated on the L.20.02 PLL Verifications screen. Then they can be assigned to verification jobs on the L.20.03 Verification Jobs screen to be executed when running a job.

FINISH_JOB
Description: Finishes a procedure (common for all).
Type: Procedure
Usage: L.20.02, L.20.03
Event: Called after a job stops processing.
Default: NULL

INIT_JOB
Description: Initializes a procedure (common for all).
Type: Procedure
Usage: L.20.02, L.20.03
Event: Called before a job starts processing.
Default: NULL
Custom Packages

**TPTA**

Description: This procedure transfers the PLL attributes to the BOM.
Type: Procedure
Usage: L.20.02, L.20.03
Event: Verification job
Default: NULL

**TP_INSULAT**

Description: This procedure generates insulation in the PLL attributes.
Type: Procedure
Usage: L.20.02, L.20.03
Event: Verification job
Default: NULL

**TP_IN_THK**

Description: This procedure updates the insulation thickness in the PLL module.
Type: Procedure
Usage: L.20.02, L.20.03
Event: Verification job
Default: NULL
**UPDATE_LP**

Description: This procedure updates the requisition indicator in the BOM position.

Type: Procedure

Usage: L.20.02, L.20.03

Event: Verification job

Default: UPDATE m_List_pos lp SET lp.requisition_ind = 'N'

**M_PCK_PO_CUSTOM**

This package contains all customizable procedures and functions for procurement (MSCM).

**AGREEMENT_APPROVAL**

Description: This CIP function is called during the approval of an agreement. The return value is of type BOOLEAN. If the return value is FALSE, the approval procedure stops; if the return value is TRUE, the approval procedure continues.

The parameter 'p_check_only_ind' is 'Y' or 'N', depending on the point within the approval procedure where this function is called.

When the RfA check box on P.50.07 is selected, this function is called with 'p_check_only_ind' set to 'Y'. When the Approve button (or the Finalize Approve button in cases where the approval sequence is used) is clicked, this function is called with 'p_check_only_ind' set to 'N'.

Type: Function

Usage: P.50.07, P.50.16

Event: Approve agreement

Default: TRUE
**BEFORE_POH_CREATION**

Description: This procedure is executed when an agreement is created. If this CIP raises an error, the agreement creation process is stopped.

Type: Procedure

Usage: OPI.10.01, P.30.01, P.30.02, P.30.04, P.50.01, P.50.07, SM.40.04

Event: Create agreement

Default: NULL

**CHECK_PO_NUMBER**

Description: This function checks the agreement number when the agreement is approved on the P.50.07 Maintain Agreements screen.

Type: Procedure

Usage: P.50.07

Event: Approve agreement

Default: NULL

**CHECK_PRINT_ORDER**

Description: This function checks whether the Print Agreement button is to be activated on the Agreement tab of the P.50.07 Maintain Agreements screen.

If the function returns NULL, the button is not enabled. If the function returns a report name, the Print Agreement button is enabled. If you click this button, the A.60.71 screen opens using this report name.

Type: Function

Usage: P.50.07

Event: Open screen

Default: NULL (sample code available)
CHECK_REV_APP_ALLOWED
Description: This function checks whether reversing of order approval is allowed.
Type: Function
Usage: P.50.07
Event: Click the Reverse Approval button.
Default: TRUE

CHECK_RFA
Description: This function checks whether the selection of the Ready for Approval check box is allowed.
If a value <> 0 is returned, an error message appears, and data is rolled back.
Type: Function
Usage: P.50.07
Event: Select the Ready for Approval check box.
Default: 0

CHECK_SET_ISSUE_DATE
Description: This function is executed before the issue date is set when you click the Issue button on the P.50.07 Maintain Agreements screen.
Type: Function
Usage: P.50.07
Event: Click the Issue button.
Default: TRUE
Custom Packages

**DEFAULT_ACCOUNT**

Description: This procedure fills the PO header with a default account code.

**IMPORTANT:** This procedure should not contain a COMMIT because the procedure execution performs the commit. (All database inserts/updates are to be performed or none at all.)

Type: Procedure

Usage: P.30.01, P.50.07

Event: Create agreement

Default: NULL

**DELETE_ORDER**

Description: This procedure is executed when an order is deleted.

**IMPORTANT:** This procedure should not contain a COMMIT because the procedure execution performs the commit. (All database inserts/updates are to be performed or none at all.)

Type: Procedure

Usage: P.50.07

Event: Delete agreement

Default: NULL

**EXECUTE_CIP**

Description: This procedure is executed when the Calculate Tax button on the Total Costs tab of the P.50.07 Maintain Orders screen is clicked.

**IMPORTANT:** This procedure should not contain a COMMIT because the procedure execution performs the commit. (All database inserts/updates are to be performed or none at all.)

Type: Procedure

Usage: P.50.07

Event: Click the Calculate Tax button.
EXEC_GENERAL_CIP

Description: This procedure is executed when the Execute CIP button on the Agreements tab of the P.50.07 Maintain Orders screen is clicked.

**IMPORTANT** This procedure should not contain a COMMIT because the procedure execution performs the commit. (All database inserts/updates are to be performed or none at all.)

Type: Procedure

Usage: P.50.07

Event: Click the Execute CIP button.

Default: NULL

GEN_INQ_NUMBER

Description: This function generates an inquiry number based on customer-specific rules. This function is called each time an inquiry (supplement) is created. If the function returns NULL, the standard default number is taken.

Type: Function

Usage: P.30.01, P.30.11

Event: Create inquiry

Default: NULL
**GEN_ORDER_NUMBER**

Description: This function generates an order number based on customer-specific rules. This function is called each time an order (supplement) is created. If the function returns NULL, the standard default number is taken.

Type: Function

Usage: P.30.01, P.30.02, P.30.04, P.50.01, P.50.07

Event: Create agreement

Default: NULL

---

**GET_PO_STATUS**

Description: This function can be used to get the agreement status, including color and priority information, and display it in a CIP field on the P.50.01 Maintain Agreements screen.

Type: Function

Usage:

Event:

Default: Status, color, and priority of the agreement

---

**GET_SUPPL_DELV_STATUS**

Description: This function identifies the delivery status of an agreement supplement.

Type: Function

Usage: MSCM

Event: View agreement

Default: -, Shipped, or Received, depending on the Shipped and Received indicators.
**GET_SUPPL_STATUS**

Description: This function identifies the status of an agreement supplement.

Type: Function

Usage: MSCM

Event: View agreement

Default: -, Ready for Approval, Approval in Progress, Approved, Shipped, Received, or Closed

**IMPORT_PB_ITEMS_CIP**

Description: The purpose of this CIP is to import items into the price breakdown list of the agreement.

Type: Procedure

Usage: Click the Import Data button on the Schedule of Values tab of P.50.07.

Event: Create agreement

Default: NULL

**POST_APPLY_PRICES**

Description: This procedure is executed after the prices have been applied from a quote summary, an agreement, or a price agreement to the given agreement.

**IMPORTANT** A commit must be performed to apply the changes to the database.

Type: Procedure

Usage: P.50.07

Event: Click the Apply Prices button.

Default: NULL
Custom Packages

POST_APPROVAL

Description: This procedure is executed after an order has been successfully approved.

IMPORTANT: This procedure must contain a COMMIT if data is updated.

Type: Procedure

Usage: P.50.07

Event: Approve agreement

Default: NULL

POST_POH_CREATION

Description: This procedure is executed after an agreement has been created using the Create Agreement button on the P.50.01 Agreements Window 4 screen. In addition, the procedure is executed when a change order (CO) is created with the Create CO button on the P.50.07, SM.40.04, and OPI screens.

You can use this CIP, for example, to modify the agreement number with information that is only available after an agreement has been created, such as the Issued by information or the associated engineering requisition to be reflected in the agreement number.

Type: Procedure

Usage: OPI.10.01, P.50.01, P.50.07, SM.40.04

Event: Create agreement

Default: NULL

POST_REVERSE_APPROVAL

Description: This procedure is executed after the approval of an order has been successfully reversed.

IMPORTANT: This procedure must contain a COMMIT if data is updated.

Type: Procedure

Usage: P.50.07

Event: Click the Reverse Approval button.
Custom Packages

Default: NULL

POST_SET_ISSUE_DATE

Description: This procedure is executed after the issue date is set when you click the Issue button on the P.50.07 Maintain Agreements screen.

IMPORTANT This procedure must contain a COMMIT if data is updated.

Type: Procedure

Usage: P.50.07

Event: Click the Issue button.

Default: NULL

RENUMBER_POS

Description: This procedure renumbers the positions of order line items when the Renumber button on the Line Items tab of the P.50.07 Maintain Agreements screen is clicked.

Type: Procedure

Usage: P.50.07

Event: Click the Renumber button.

Default: Line item renumbering starting with 1, increment 1

M_PCK_QUERY_CUSTOM

This package contains all customizable procedures and functions that are needed to maintain user-defined views (A.60.06) in the case of linked tables.
Custom Packages

CHECK_COND

Description: Function for checking the condition when the master table is used.

Type: Function

Usage: A.60.06

Event: Create Where Condition

Default: see source code

M_PCK_QUOTE_CUSTOM

This package contains all customizable procedures and functions for quotes.

AFTER_CHECK_INQ_PRECON

Description: This CIP is called after preconditions of an inquiry are checked when the RfA check box is selected on P.30.21 Prepare RFQ.

Type: Procedure

Usage: P.30.21

Event: Selection of inquiry RfA check box

Default: NULL

BEFORE_APPROVAL

Description: This procedure is called before the approval of the quote summary in m_pck_qs.approve_qs.

The procedure does appropriate checks and raises an error if necessary. This error will be passed to the screen.

Type: Procedure

Usage: P.30.22

Event: Quote Summary Approval

Default: NULL
CHECK_BEFORE_INQ_APPR

Description: This procedure is called before the approval of an inquiry. In this CIP, you can use the entry of the Justification field on the Inquiry tab of P.30.21 Prepare RFQ to determine whether the approval can be done or not. For example, if the Justification field contains any text, the inquiry can be approved.

Type: Procedure

Usage: P.30.21

Event: Inquiry Approval

Default: NULL

GET_ALT_PROMPT

Description: Function for setting the label of the Alt check box on the Awarding tab.

Type: Function

Usage: P.30.25

Event: Open screen

Default: ‘Alt’

GET_CE_MISSING_PROMPT

Description: Function for setting the label of the CE Missing check box on the Awarding tab.

Type: Function

Usage: P.30.25

Event: Open screen

Default: ‘CE Missing’
GET_COMMODITY_CODE_PROMPT

Description: Function for setting the label of the Commodity Code field on the Awarding tab.
Type: Function
Usage: P.30.25
Event: Open screen
Default: ‘Commodity Code’

GET_EIGHTH_AWARDING_VALUE

Description: Function for getting a value for the eighth variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Required delivery date of alternate 0 of a quote detail. If an alternate has been selected, the required delivery date of that alternate is returned instead.

GET_EIGHTH_PROMPT

Description: Function for setting the label of the eighth variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: ‘Delivery Date’
GET_EIGHTH_VALUE
Description: Function for getting a value for the eighth variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Required delivery date of alternate 0 of a quote detail. If an alternate has been selected, the required delivery date of that alternate is returned instead.

GET_FIFTH_AWARDING_SUM
Description: Function for getting a value for the fifth matrix summary line field. Return value is of data type 'Number' and can have up to 26 digits. This function is not called/used for virtual bidders.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Price index

GET_FIFTH_AWARDING_VALUE
Description: Function for getting a value for the fifth variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Unit price of alternate 0. If an alternate has been selected, the unit price of that alternate is returned instead.
**GET_FIFTH_FORMAT**

Description: Function for getting a format mask for the fifth variable matrix body field.

Type: Function

Usage: P.30.25

Event: Open screen

Default: '999g999g990d9999'

**GET_FIFTH_PROMPT**

Description: Function for setting the label of the fifth variable matrix body field.

Type: Function

Usage: P.30.25

Event: Open screen

Default: 'Unit Price'

**GET_FIFTH_SUM_LINE**

Description: Function for getting a value for the fifth matrix summary line field. Return value is of data type 'Number' and can have up to 26 digits. This function is also called/used for virtual bidders and tender reference amounts.

Type: Function

Usage: P.30.25

Event: Open screen

Default: Price index
GET_FIFTH_SUM_LINE_FORMAT
Description: Function for getting a format mask for the fifth matrix summary line field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: '999G999G999G999G999G990D00'

GET_FIFTH_SUM_LINE_PROMPT
Description: Function for setting the label of the fifth matrix summary line field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: 'Price Index'

GET_FIFTH_VALUE
Description: Function for getting a value for the fifth variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Unit price of alternate 0. If an alternate has been selected, the unit price of that alternate is returned instead.
Custom Packages

GET_FIRST_AWARDING_SUM
Description: Function for getting a value for the first matrix summary line field. Return value is of data type 'Number' and can have up to 26 digits. This function is not called/used for virtual bidders.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Total other costs assigned on the quote summary level (excluding evaluation costs).

GET_FIRST_AWARDING_VALUE
Description: Function for getting a value for the first variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Quoted quantity of alternate 0. If an alternate has been selected, the quoted quantity of that alternate is returned instead.

GET_FIRST_FORMAT
Description: Function for getting a format mask for the first variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: '999g999g990d9999'
GET_FIRST_PROMPT

Description: Function for setting the label of the first variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: 'Quoted Quantity'

GET_FIRST_SUM_LINE

Description: Function for getting a value for the first matrix summary line field. Return value is of data type 'Number' and can have up to 26 digits. This function is also called/used for virtual bidders and tender reference amounts.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Total other costs assigned on the quote summary level (excluding evaluation costs).

GET_FIRST_SUM_LINE_FORMAT

Description: Function for getting format mask of the first matrix summary line field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: '999G999G999G999G999G990D00'
Custom Packages

**GET_FIRST_SUM_LINE_PROMPT**

Description: Function for setting the label of the first matrix summary line field.

Type: Function

Usage: P.30.25

Event: Open screen

Default: 'Other Costs QS'

**GET_FIRST_TOTAL_FORMAT**

Description: Function for getting format mask of the first total field.

Type: Function

Usage: P.30.25

Event: Open screen

Default: '999G999G999G999G999G990D00'

**GET_FIRST_TOTAL_PROMPT**

Description: Function for setting the label of the first total field.

Type: Function

Usage: P.30.25

Event: Open screen

Default: 'Total OC'
GET_FIRST_TOTAL_VALUE
Description: Function for getting value for the first total field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Total sum of other costs

GET_FIRST_VALUE
Description: Function for getting value for the first variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Quoted quantity of alternate 0. If an alternate has been selected, the quoted quantity of that alternate is returned instead.

GET_FIRST_VB_HEADER
Description: Function for getting value of the first variable matrix header field for a virtual bidder.
Type: Function
Usage: P.30.25
Event: Open screen
Default: 'Virtual Bidder', repeated if ZP_CE_RPV is set to Y
**GET_FOURTH_AWARDING_SUM**

**Description:** Function for getting value for the first matrix summary line field. Return value is of data type 'Number' and can have up to 26 digits. This function is not called/used for virtual bidders.

**Type:** Function

**Usage:** P.30.25

**Event:** Open screen

**Default:** Total costs for all quote details, including all packing costs and other costs (of both the summary and detail level). Alternate/option costs and addenda costs are also included.

---

**GET_FOURTH_AWARDING_VALUE**

**Description:** Function for getting value for the fourth variable matrix body field.

**Type:** Function

**Usage:** P.30.25

**Event:** Open screen

**Default:** Basic costs for a quote detail of alternate 0.

---

**GET_FOURTH_FORMAT**

**Description:** Function for getting format mask of the fourth variable matrix body field.

**Type:** Function

**Usage:** P.30.25

**Event:** Open screen

**Default:** '999G999G999G999G999G990D00'
GET_FOURTH_PROMPT

Description: Function for setting the label of the fourth variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: 'Basic Costs'

GET_FOURTH_SUM_LINE

Description: Function for getting value for the fourth matrix summary line field. Return value is of data type 'Number' and can have up to 26 digits. This function is also called/used for virtual bidders and tender reference amounts.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Total costs for all quote details, including all packing costs and other costs (of both the summary and detail level). Alternate/option costs and addenda costs are also included.

GET_FOURTH_SUM_LINE_FORMAT

Description: Function for getting format mask of the fourth matrix summary line field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: '999G999G999G999G999G990D00'
Custom Packages

GET_FOURTH_SUM_LINE_PROMPT
Description: Function for setting the label of the fourth matrix summary line field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: ‘All Pos + OC + Alt/Opt + Add’

GET_FOURTH_VALUE
Description: Function for getting value for the fourth variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Basic costs for a quote detail of alternate 0.

GET_GROUP_BY_1_PROMPT
Description: Function for setting the label of the Group by 1 field on the Awarding tab.
Type: Function
Usage: P.30.25
Event: Open screen
Default: ‘Group by 1’
GET_GROUP_BY_2_PROMPT
Description: Function for setting the label of the Group by 2 field on the Awarding tab.
Type: Function
Usage: P.30.25
Event: Open screen
Default: 'Group by 2'

GET_GROUP_BY_3_PROMPT
Description: Function for setting the label of the Group by 3 field on the Awarding tab.
Type: Function
Usage: P.30.25
Event: Open screen
Default: 'Group by 3'

GET_GROUP_BY_4_PROMPT
Description: Function for setting the label of the Group by 4 field on the Awarding tab.
Type: Function
Usage: P.30.25
Event: Open screen
Default: 'Group by 4'
Custom Packages

GET_GROUP_BY_5_PROMPT
Description: Function for setting the label of the Group by 5 field on the Awarding tab.
Type: Function
Usage: P.30.25
Event: Open screen
Default: 'Group by 5'

GET_IDENT_PROMPT
Description: Function for setting the label of the Ident Code field on the Awarding tab.
Type: Function
Usage: P.30.25
Event: Open screen
Default: 'Ident Code'

GET_IDENT_SHORT_PROMPT
Description: Function for setting the label of the Ident Short Description field on the Awarding tab.
Type: Function
Usage: P.30.25
Event: Open screen
Default: 'Ident Short Description'
GET_LINE_MARK
Description: Function for setting color visualization on the Awarding tab.
Type: Function
Usage: P.30.25
Event: Open screen
Default: 0 (Sample code available)

GET_NINTH_AWARDING_VALUE
Description: Function for getting value for the ninth variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Unit price type of alternate 0 of a quote detail is displayed. If an alternate has been selected, the unit price type of that alternate is returned instead.

GET_NINTH_PROMPT
Description: Function for setting the label of the ninth variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Unit Price Type
**GET_NINTH_VALUE**
Description: Function for getting value for the ninth variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Unit price type of alternate 0 of a quote detail is displayed. If an alternate has been selected, the unit price type of that alternate is returned instead.

**GET_NO_BID_PROMPT**
Description: Function for setting the label of the No Bid check box on the Awarding tab.
Type: Function
Usage: P.30.25
Event: Open screen
Default: 'No Bid'

**GET_ORIGIN_PROMPT**
Description: Function for setting the label of the Origin field on the Awarding tab.
Type: Function
Usage: P.30.25
Event: Open screen
Default: 'Origin'
GET_POS_PROMPT
Description: Function for setting the label of the Pos (position number of requisition line item) field on the Awarding tab.
Type: Function
Usage: P.30.25
Event: Open screen
Default: ‘Pos’

GET_PROC_SEQ_PROMPT
Description: Function for setting the label of the Proc Seq field on the Awarding tab.
Type: Function
Usage: P.30.25
Event: Open screen
Default: ‘Proc Seq’

GET_R_CODE_PROMPT
Description: Function for setting the label of the Engineering Requisition field on the Awarding tab.
Type: Function
Usage: P.30.25
Event: Open screen
Default: ‘Engineering Requisition’
Custom Packages

GET_R_SUPP_PROMPT

Description: Function for setting the label of the Suppl field on the Awarding tab.
Type: Function
Usage: P.30.25
Event: Open screen
Default: 'Suppl'

GET_SECOND_AWARDING_SUM

Description: Function for getting value for the second matrix summary line field. Return value is of data type 'Number' and can have up to 26 digits. This function is not called/used for virtual bidders.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Total costs for all selected quote details are displayed. These total costs also include the packing price and other costs assigned on the quote detail level.

GET_SECOND_AWARDING_VALUE

Description: Function for getting value for the second variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Total costs of the quote detail, alternate 0, including other costs, packing price, and discount amount. If an alternate has been selected, the total costs of that alternate are returned instead.
**GET_SECOND_FORMAT**

Description: Function for getting format mask of the second variable matrix body field.

Type: Function

Usage: P.30.25

Event: Open screen

Default: '999G999G999G999G999G990D00'

**GET_SECOND_PROMPT**

Description: Function for setting the label of the second variable matrix body field.

Type: Function

Usage: P.30.25

Event: Open screen

Default: 'Full Price'

**GET_SECOND_SUM_LINE**

Description: Function for getting value for the second matrix summary line field. Return value is of data type 'Number' and can have up to 26 digits. This function is also called/used for virtual bidders and tender reference amounts.

Type: Function

Usage: P.30.25

Event: Open screen

Default: Total costs for all selected quote details are displayed. These total costs also include the packing price and other costs assigned on the quote detail level.
**GET_SECOND_SUM_LINE_FORMAT**

Description: Function for getting format mask of the second matrix summary line field.

Type: Function

Usage: P.30.25

Event: Open screen

Default: '999G999G999G999G999G990D00'

**GET_SECOND_SUM_LINE_PROMPT**

Description: Function for setting the label of the second matrix summary line field.

Type: Function

Usage: P.30.25

Event: Open screen

Default: 'Selected Pos'

**GET_SECOND_TOTAL_FORMAT**

Description: Function for getting format mask of the second total field.

Type: Function

Usage: P.30.25

Event: Open screen

Default: '999G999G999G999G999G990D00'
CUSTOM PACKAGES

GET_SECOND_TOTAL_PROMPT
Description: Function for setting the label of the second total field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: 'Total LI Costs'

GET_SECOND_TOTAL_VALUE
Description: Function for getting value for the second total field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Total line item costs

GET_SECOND_VALUE
Description: Function for getting value for the second variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Total costs of the quote detail, alternate 0, including other costs, packing price, and discount amount. If an alternate has been selected, the total costs of that alternate are returned instead.
GET_SECOND_VB_HEADER
Description: Function for getting value of the second variable matrix header field for virtual bidders.
Type: Function
Usage: P.30.25
Event: Open screen
Default: 'Virtual Bidder'

GET_SEVENTH_AWARDING_VALUE
Description: Function for getting value for the seventh variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Discount amount of alternate 0. If an alternate has been selected, the discount amount of that alternate is returned instead.

GET_SEVENTH_FORMAT
Description: Function for getting format mask of the seventh variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: '999g999g990d9999'
GET_SEVENTH_PROMPT
Description: Function for setting the label of the seventh variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: ‘Discount Amount’

GET_SEVENTH_VALUE
Description: Function for getting value for the seventh variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Discount amount of alternate 0 is shown. If an alternate has been selected, the discount amount of that alternate is returned instead.

GET_SIXTH_AWARDING_SUM
Description: Function for getting value for the sixth matrix summary line field. Return value is of data type ‘Character’ and can have up to 100 digits. This function is not called/used for virtual bidders.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Payment terms
GET_SIXTH_AWARDING_VALUE

Description: Function for getting value for the sixth variable matrix body field.

Type: Function

Usage: P.30.25

Event: Open screen

Default: Packing price of alternate 0. If an alternate has been selected, the packing price of that alternate is returned instead.

GET_SIXTH_FORMAT

Description: Function for getting format mask of the sixth variable matrix body field.

Type: Function

Usage: P.30.25

Event: Open screen

Default: '999g999g990d9999'

GET_SIXTH_PROMPT

Description: Function for setting the label of the sixth variable matrix body field.

Type: Function

Usage: P.30.25

Event: Open screen

Default: ‘Packing Price’
GET_SIXTH_SUM_LINE

Description: Function for getting value for the sixth matrix summary line field. Return value is of data type 'Character' and can have up to 100 digits. This function is also called/used for virtual bidders.

Type: Function
Usage: P.30.25
Event: Open screen
Default: m_pck_qs.get_payment_term(p_inq_id, p_index)

GET_SIXTH_SUM_LINE_PROMPT

Description: Function for setting the label of the sixth matrix summary line field.

Type: Function
Usage: P.30.25
Event: Open screen
Default: 'Payment Terms'

GET_SIXTH_VALUE

Description: Function for getting value for the sixth variable matrix body field.

Type: Function
Usage: P.30.25
Event: Open screen
Default: Packing price of alternate 0 is displayed. If an alternate has been selected, the packing price of that alternate is shown instead.
Custom Packages

**GET_SIZE_1_PROMPT**
- **Description:** Function for setting the label of the Size 1 field on the Awarding tab.
- **Type:** Function
- **Usage:** P.30.25
- **Event:** Open screen
- **Default:** ‘Size 1’

**GET_SIZE_2_PROMPT**
- **Description:** Function for setting the label of the Size 2 field on the Awarding tab.
- **Type:** Function
- **Usage:** P.30.25
- **Event:** Open screen
- **Default:** ‘Size 2’

**GET_SIZE_3_PROMPT**
- **Description:** Function for setting the label of the Size 3 field on the Awarding tab.
- **Type:** Function
- **Usage:** P.30.25
- **Event:** Open screen
- **Default:** ‘Size 3’
GET_SIZE_4_PROMPT
Description: Function for setting the label of the Size 4 field on the Awarding tab.
Type: Function
Usage: P.30.25
Event: Open screen
Default: ‘Size 4’

GET_SIZE_5_PROMPT
Description: Function for setting the label of the Size 5 field on the Awarding tab.
Type: Function
Usage: P.30.25
Event: Open screen
Default: ‘Size 5’

GET_SPLIT_PROMPT
Description: Function for setting the label of the Split check box on the Awarding tab.
Type: Function
Usage: P.30.25
Event: Open screen
Default: ‘Split’
**GET_SUB_POS_PROMPT**

Description: Function for setting the label of the **Sub** (sub position number of requisition line item) field on the **Awarding** tab.

Type: Function

Usage: P.30.25

Event: Open screen

Default: ‘Sub’

**GET_SUM_LINE_MARK**

Description: Function for setting color visualization on the **Awarding** tab.

Type: Function

Usage: P.30.25

Event: Open screen

Default: 0 (sample code available)

**GET_TAG_NUMBER_PROMPT**

Description: Function for setting the label of the **Tag Number** field on the **Awarding** tab.

Type: Function

Usage: P.30.25

Event: Open screen

Default: ‘Tag Number’
GET_TE_MISSING_PROMPT
Description: Function for setting the label of the TE Missing check box on the Awarding tab.
Type: Function
Usage: P.30.25
Event: Open screen
Default: ‘TE Missing’

GET_THIRD_AWARDING_SUM
Description: Function for getting value for the third matrix summary line field. Return value is of data type ‘Number’ and can have up to 26 digits. This function is not called/used for virtual bidders.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Total costs for all selected quote details plus the total other costs assigned on the quote summary level.

GET_THIRD_AWARDING_VALUE
Description: Function for getting value for the third variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Sum of the other costs on the quote detail level for alternate 0. Other costs that have been marked as evaluation costs are not included in this total.
**GET_THIRD_FORMAT**

Description: Function for getting format mask of the third variable matrix body field.

Type: Function

Usage: P.30.25

Event: Open screen

Default: '999G999G999G999G999G990D00'

**GET_THIRD_PROMPT**

Description: Function for setting the label of the third variable matrix body field.

Type: Function

Usage: P.30.25

Event: Open screen

Default: 'Other Costs'

**GET_THIRD_SUM_LINE**

Description: Function for getting value for the third matrix summary line field. Return value is of data type 'Number' and can have up to 26 digits. This function is also called/used for virtual bidders and tender reference amounts.

Type: Function

Usage: P.30.25

Event: Open screen

Default: Total costs for all selected quote details plus the total other costs assigned on the quote summary level.
**GET_THIRD_SUM_LINE_FORMAT**

Description: Function for getting format mask of the third matrix summary line field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: '999G999G999G999G999G990D00'

**GET_THIRD_SUM_LINE_PROMPT**

Description: Function for setting the label of the third matrix summary line field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: 'Selected Pos + OC'

**GET_THIRD_TOTAL_FORMAT**

Description: Function for getting format mask of the third total field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: '999G999G999G999G999G990D00'
GET_THIRD_TOTAL_PROMPT

Description: Function for setting the label of the third total field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: 'Total Best Splits'

GET_THIRD_TOTAL_VALUE

Description: Function for getting value for the third total field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Total sum of costs when applying best splits ("cherry picking").

GET_THIRD_VALUE

Description: Function for getting value for the third variable matrix body field.
Type: Function
Usage: P.30.25
Event: Open screen
Default: Sum of the other costs on the quote detail level for alternate 0. Other costs that have been marked as evaluation costs are not included in this total.
GET_THIRD_VB_HEADER
Description: Function for getting value of the third variable matrix header field for virtual bidders.
Type: Function
Usage: P.30.25
Event: Open screen
Default: 'Virtual Bidder'

IMPORT_PB_ITEMS_CIP
Description: The purpose of this CIP is to import items into the price breakdown.
Type: Procedure
Usage: P.30.21
Event: Create quote (approve inquiry)
Default: NULL

PRE_COMMIT_CREATE_INQ
Description: This CIP is executed before COMMIT is performed when an inquiry is created.
Type: Procedure
Usage: P.30.21
Event: Commit after inquiry is created
Default: NULL
**M_PCK_REPORT_CUSTOM**

Creates a generic report file.

**SET_PRINTOUT**

Description: This procedure creates a file name for your print file. Allocate this file name to variable MY_FILE_NAME. If you want this file name to be used, keep the default for variable use_file_name (Y).

Type: Procedure

Usage: A.60.71

Event: Run report

Default: example code that must be modified

**M_PCK_REQ_BLR_CUSTOM**

This package contains all customizable procedures and functions for balance requisitions.

**ALLOW_REQ_DELETION**

Description: The purpose of this CIP is to verify if deleting a balance requisition is allowed. If the deletion is allowed, then the function returns 1; otherwise, the function returns 0. This function is called by m_pck_req.del_req.

Type: Function

Usage: R.40.11

Event: Delete requisition

Default: 1
**APPROVE_REQ**

Description: This procedure is called by m_pck_req.approve_req. This procedure is called when a requisition is approved on R.30.01. This procedure is also called when ZR_ADV_APP is set to N and a requisition is released. It is possible to unapprove a requisition, change it in some way, and then approve it again. So this CIP procedure must handle being called multiple times for the same requisition.

You can implement any approval checks necessary here; you have two ways to indicate an error to the caller in order to stop the approval process:

1. set mpck_login.warning_ind to 'Y'. The caller will then RAISE_APPLICATION_ERROR(-20000,'MAR-25026'), which means, "ERROR: Cannot approve req due to failed custom approval check"

   OR

2. RAISE_APPLICATION_ERROR yourself from within approve_req. The caller will propagate this exception to the GUI. Please make sure to enter a correct custom error message text on A.60.31 if you want to use this method.

Type: Procedure

Usage: R.30.01

Event: Approve requisition

Default: NULL

---

**CALC_BUDGET**

Description: This procedure is called by m_pck_mto_blr.call_cips.

Type: Procedure

Usage: R.20.01, R.30.01

Event: MTO Job

Default: m_pck_m.ml('CIP Calculating budget ' || 'is empty', 80, FALSE, FALSE)
CHECK_FORCE_RTP

Description: This function checks whether there are any requisition line items that will not be released to procurement because there has been no change in quantity and the Force RTP check box has not been set, although changes have occurred to either attached attributes, documents, or VDRs.

Type: Function
Usage: R.30.01
Event: Release to Procurement
Default: m_pck_req_custom.check_force_rtp(p_r_id)

CREATE_PO_REQ_CIP

Description: This procedure is called when you click the Create PO Req button on R.30.01.

Type: Procedure
Usage: R.30.01
Event: Click the Create PO Req button.
Default: NULL

DESIGNATION

Description: This procedure is called by m_pck_mto_blr.call_cips.

Type: Procedure
Usage: R.20.01, R.30.01
Event: MTO Job
Default: m_pck_req_custom.designation(p_r_id)
**GENERAL_CIP**

Description: This procedure is called by m_pck_mto_blr.process_reqs. This general, all-purpose CIP has been incorporated into this package in order to allow customers to embed any kind of functionality that they may require into MTO job execution. This procedure is executed by m_pck_mto_blr, if on R.20.01, the Execute CIP check box is selected.

This code is not maintained by Intergraph; as with all CIP code, it is the customer’s responsibility to use this API and maintain their custom code. Customers must save their code as text files, because SmartPlant Reference Data and SmartPlant Materials DB migration overwrites it without any warning.

**IMPORTANT:** This procedure must not commit. Otherwise, the MTO job may not work properly.

Type: Procedure

Usage: R.20.01, R.30.01

Event: MTO Job

Default: NULL

---

**GET_IDENT_INQUIRY_QTY**

Description: This function returns the inquiry qty (user-defined).

Type: Function

Usage: C.40.R.01

Event: Run report

Default: m_pck_req_custom.get_ident_inquiry_qty(p_ident,p_unit_id,p_conv)
CUSTOM PACKAGES

GET_INQUIRY_QTY

Description: This function returns the inquiry quantity instead of m_req_line_items.inquiry_qty.

Type: Function

Usage: R.30.08, C.40.01 Window 3

Event: Open screen

Default: m_pck_req_custom.get_inquiry_qty(p_rli_id)

IMPORT_PB_ITEMS_CIP

Description: The purpose of this CIP is to import items into the price breakdown list of the requisition specified.

Type: Procedure

Usage: R.30.01

Event:

Default: NULL

INSERT_M_REQS

Description: This procedure is called by m_pck_rel_to_proc.rel_to_proc for a balance requisition.

Type: Procedure

Usage: R.30.01

Event: Release to Procurement

Default: m_pck_m.ml('CIP insert_m_reqs ' || 'is empty', 80, FALSE, FALSE)
NEW_SUPPLEMENT_CIP

Description: This procedure is called when you click the New Supplement button on R.30.01.

Type: Procedure

Usage: R.30.01

Event: Click the New Supplement button.

Default: NULL

RENUMBER_POS_SUB

Description: This procedure is executed from the Line Items tab of R.30.01 Maintain Requisitions Window 2. This procedure is called when you click the Renumber button. The renumbering is done as follows:

1. MTO line items - Commodity Code and Sizes (1,2) if group by criteria exists, then same pos and count up sub pos for a group
2. Manual line items - Commodity Code and Sizes (1,2)

Type: Procedure

Usage: R.30.01

Event: Click the Renumber button.

Default: m_pck_req_custom.renumber_pos_sub(p_r_id)

SET_FORCE_RTP

Description: This procedure sets the Force RTP check box for all requisition line items that will not be released to procurement because there has been no change in quantity and the Force RTP indicator has not been set, although changes have occurred to either attached attributes, documents, or VDRs. This procedure is called by R.30.01 when a requisition is released to procurement.

Type: Procedure

Usage: R.30.01
Event: Release requisition to procurement
Default: m_pck_req_custom.set_force_rtp(p_r_id)

**SET_PROC_ORDER_SEQ**

Description: This procedure is called by m_pck_mto_blr.process_reqs. The procedure is called if project default ZR_PROCSEQ = "Y". The procedure allows tracking of related PO lines throughout several requisition supplements by assigning order sequence numbers to requisition line items depending on supplement, increased quantity, and requisition template order criteria.

Type: Procedure
Usage: P.30.22, R.20.01, R.30.01
Event: MTO Job
Default: m_pck_req_custom.set_proc_order_seq(p_r_id)

**SET_REV_ATTR**

Description: This procedure is called by procedure m_pck_req_revision.rev_exec_main. This CIP has been incorporated into this package in order to allow customers to set values for req attributes after inserting a record in m_reqs.

- Reqs created manually have an RTP_ID < 0
- Reqs created by ORI have an RTP_ID = 0
- Reqs created by MTO have the MTY_ID <> 0

Type: Procedure
Usage: R.20.01, R.30.01
Event: Create requisition
Default: NULL
**STD_JOB_ID**

Description: This procedure is called by m_pck_mto_blr.process_reqs.

Type: Procedure

Usage: R.20.01, R.30.01

Event: MTO Job

Default: m_pck_m.ml('CIP Getting account ' || 'is empty', 30, FALSE, FALSE)

**STD_JOB_INIT**

Description: This procedure initializes an MTO job. The procedure is called by M_PCK_MTO.INIT_JOB.

Type: Procedure

Usage: R.20.01

Event: MTO Job

Default: NULL

**STD_PROC_CHANNEL**

Description: This procedure is called by m_pck_mto_blr.call_cips.

Type: Procedure

Usage: R.20.01, R.30.01

Event: MTO Job

Default: m_pck_m.ml('CIP Getting procure channel ' || 'is empty', 80, FALSE, FALSE)
Custom Packages

STD_REVERSE_REL
Description: This procedure is called by m_pck_rel_to_proc.rev_rel_to_proc.
Type: Procedure
Usage: R.40.11
Event: Update requisition status
Default: NULL

STD_ROS_DATE
Description: This procedure is called by m_pck_mto_blr.call_cips.
Type: Procedure
Usage: R.20.01, R.30.01
Event: MTO Job
Default: Estimated ROS Date

SUM_INQUIRY
Description: This procedure returns values for the inquiry quantity, instead of m_pck_ident_sums.sum_inquiry returning the values.
Type: Procedure
Usage: C.40.01
Event: Query ident
Default: m_pck_req_custom.sum_inquiry(p_ident,p_unit_id,p_sum_field,p_sum_without_unit,p_tag_number)
TECH_EVAL

Description: This procedure is called by m_pck_mto_blr.call_cips.

Type: Procedure

Usage: R.20.01

Event: MTO Job

Default: m_pck_m.ml('CIP Setting tech eval indicator ' || ' is empty', 80, FALSE, FALSE)

UPDATE_INQ_REQ

Description: This procedure is called by m_pck_mto_blr.update_inq_req. This procedure is called when an inquiry only requisition is revised (that is, the original supplement 0 is updated instead of creating a new supplement).

p_temp_new_suppl_r_id is the internal primary key of that supplement 1 requisition which is the source of the update but exists only temporarily, meaning that after this CIP has been called, m_pck_mto.update_inq_req deletes this temporary supplement and unapproves ("opens") supplement 0 again.

p_original_inq_req_r_id is the internal PK of the revised (that is, suppl 0) inquiry requisition.

When this CIP is called, all updates to supplement 0 have already been made.

Possible applications of this CIP are, for example:

- calling other API routines to external interfaces that must be informed about this internal event
- reconciling any attachment data from supplement 1 to supplement 0 that is not done automatically by standard MTO functionality, and so on.

Type: Procedure

Usage: R.20.01

Event: MTO Job

Default: NULL
**M_PCK_REQ_CUSTOM**

This package contains all customizable procedures and functions of requisitions.

**ALLOW_REQ_DELETION**

Description: The purpose of this CIP is to verify if deleting a balance requisition is allowed. If the deletion is allowed, then the function returns 1; otherwise, the function returns 0. This function is called by m_pck_req.del_req.

Type: Function

Usage: R.40.11

Event: Delete Requisition

Default: 1

**APPROVE_RELEASE_REQ**

Description: This procedure is called by m_pck_req_load.ins_upd_req. The procedure is determines whether to approve/release the requisition.

Type: Procedure

Usage: R.30.01

Event: Approve requisition

Default: NULL
**APPROVE_REQ**

Description: This procedure is called by m_pck_req.approve_req. This procedure is executed when a requisition is approved on R.30.01. This procedure is also executed when ZR_ADV_APP is set to N and a requisition is released. It is possible to unapprove a requisition, change it in some way, and then approve it again. So this CIP handles being called multiple times for the same requisition.

You can implement any approval checks necessary here; you have two ways to indicate an error to the caller in order to stop the approval process:

1. set mpck_login.warning_ind to 'Y'. The caller will then RAISE_APPLICATION_ERROR(-20000,'MAR-25026'), meaning: "ERROR: Cannot approve req due to failed custom approval check"

or

2. RAISE_APPLICATION_ERROR yourself from within approve_req. The caller will propagate this exception to the GUI. Please make sure to enter a correct custom error message text on A.60.31 if you want to use this method.

Type: Procedure

Usage: A.20.49, R.30.01, Web Approval

Event: Requisition approval

Default: NULL

**APPROVE_REQ_BY**

Description: This procedure is called when the Approved indicator is selected on A.20.49. It is possible to revoke approval and then set the Approved indicator again. So this CIP handles being called multiple times for the same requisition and approver.

You can implement any approval checks necessary here. You can indicate an error to the caller in order to stop the approval process in the following way:

RAISE_APPLICATION_ERROR yourself from within approve_req_by. The caller will propagate this exception to the GUI. Please make sure to enter a correct custom error message text on A.60.31 if you want to use this method.

Type: Procedure

Usage: A.20.49, Web Approval

Event: Select the Approved check box.
**ATTACH_BOM_DOC_REQ_LINES_NODES**

Description: This procedure is called by M_PCK_MTO. This procedure attaches documents on BOM nodes to requisitions and on BOM positions to requisition lines.

Type: Procedure

Usage: R.20.01

Event: MTO Job

Default: See CIP source code

**CALC_BUDGET**

Description: This procedure is called by m_pck_mto.process_reqs.

Type: Procedure

Usage: R.20.01, R.30.01

Event: MTO Job

Default: NULL

**CHECK_FORCE_RTP**

Description: This function checks whether there are any requisition line items that will not be released to procurement because there has been no change in quantity and the Force RTP check box has not been set, although changes have occurred to attached attributes, documents, or VDRs.

Type: Function

Usage: R.30.01

Event: Release to Procurement

Default: See CIP source code
CHECK_RFA

Description: This function checks whether the Ready for Approval check box can be set. This function is called from R.30.01 when the Ready for Approval check box is selected.

This allows you to control when the requisition is completed and the approval process can be started.

If the Ready for Approval check box can be set, the function returns 0, which means the selection of the check box is accepted. Otherwise, the function returns a nonzero value and p_error_msg contains a short description of the error. The error description is shown to the user and the check box is cleared.

**NOTE** Only the first 120 characters stored in p_error_msg are displayed.

Type: Function
Usage: R.30.01
Event: Select the Ready for Approval check box.
Default: 0

CREATE_PO_REQ_CIP

Description: This procedure is called when you click the Create PO Req button on R.30.01.

Type: Procedure
Usage: R.30.01
Event: Click the Create PO Req button.
Default: NULL
DESIGNATION

Description: This procedure is called by m_pck_mto.process_reqs.
Type: Procedure
Usage: R.20.01, R.30.01
Event: MTO Job
Default: UPDATE m_req_line_items SET dd_id = m_pck_pd_zp.zp_dlv_dsg

GENERAL_CIP

Description: This procedure is called by m_pck_mto.process_reqs. This general, all-purpose CIP has been incorporated into this package in order to allow customers to embed any kind of functionality that they may require into MTO job execution. This procedure is executed by m_pck_mto, if on R.20.01, the Execute CIP check box is checked.

IMPORTANT: This procedure must not commit. Otherwise, the MTO job may not work properly.
Type: Procedure
Usage: R.20.01, R.30.01
Event: MTO Job
Default: NULL

GET_IDENT_INQUIRY_QTY

Description: This function returns the inquiry qty (user-defined).
Type: Function
Usage: C.40.R.01
Event: Run report
Default: 0
**GET_INQUIRY_QTY**

Description: This function returns the inquiry quantity, instead of `m_req_line_items.inquiry_qty` returning the quantities.

Type: Function

Usage: R.30.08, C.40.01 Window 3

Event: Query ident

Default: 0

**GET_ORDER**

Description: This function retrieves the BOM position order.

Type: Function

Usage:

Event:

Default: `SELECT MIN(m_pck_sort_fncts.sort_char(lp_pos)) FROM m_list_pos`

**IMPORT_PB_ITEMS_CIP**

Description: The purpose of this CIP is to import items into the price breakdown list of the requisition specified.

Type: Procedure

Usage: R.30.01

Event:

Default: NULL
**INSERT_M_REQS**

Description: This procedure is called by m_pck_rel_to_proc.rel_to_proc.

Type: Procedure

Usage: R.20.01, R.30.01, SAP.50.01

Event: Release to Procurement

Default: NULL

**NEW_SUPPLEMENT_CIP**

Description: This procedure is called when you click the New Supplement button on R.30.01.

Type: Procedure

Usage: R.30.01

Event: Click the New Supplement button.

Default: NULL

**ORDER_REQ**

Description: This procedure orders the requisition line items.

Type: Procedure

Usage: R.20.01

Event: MTO Job

Default: See CIP source code
**QTY_UNIT**

Description: This function returns the ident quantity unit.

Type: Function

Usage: AS.10.02, B.20.01.20, C.10.19, C.40.01, C.50.31, P.50.07, P.50.16, R.20.01, R.30.01, S.80.21

Event:

Default: Qty unit of the object parameter

**RENUMBER_POS_SUB**

Description: This procedure is called when you click the Renumber button on R.30.01 Maintain Requisitions. The renumbering is done as follows:

1. MTO line items
   Commodity Code and Sizes (1,2) if group by criteria exists, then same pos and count up sub pos for a group

2. Manual line items
   Commodity Code and Sizes (1,2)

Type: Procedure

Usage: R.30.01

Event: Click the Renumber button.

Default: See CIP source code

**SET_FORCE_RTP**

Description: This procedure sets the Force RTP check box for all requisition line items that will not be released to procurement because there has been no change in quantity and the Force RTP indicator has not been set, although changes have occurred to attached attributes, documents, or VDRs. This procedure is called by R.30.01 when a requisition is released to procurement.

Type: Procedure

Usage: R.30.01
Custom Packages

Event: Release requisition to procurement
Default: See CIP source code

**SET_PROC_ORDER_SEQ**

Description: This procedure is called by m_pck_mto.process_reqs. The procedure is executed if project default ZR_PROCSEQ is set to Y. It allows tracking of related PO lines throughout several requisition supplements by assigning order sequence numbers to requisition line items depending on supplement, increased quantity, and requisition template order criteria.

Type: Procedure
Usage: P.30.22, R.20.01, R.30.01
Event: MTO Job
Default: See CIP source code

**SET_REV_ATTR**

Description: This procedure is called by m_pck_req_revision.rev_exec_main. This CIP has been incorporated into this package in order to allow customers to set values for req attributes after insertion of a record in m_reqs.

- Reqs created manually have an RTP_ID < 0,
- Reqs created by ORI have an RTP_ID = 0,
- Reqs created by MTO have the MTY_ID <> 0.

**Important:** This procedure must not commit. Otherwise, the requisition revision may not work properly.

Type: Procedure
Usage: R.20.01, R.30.01
Event: Create requisition
Default: NULL
Custom Packages

**STD\_DISPO\_QTY**

Description: This function returns the dispo quantity of a requisition line item.

Type: Function

Usage: R.20.01

Event: MTO Job

Default: See CIP source code

**STD\_GLOBAL\_DISPO\_QTY**

Description: This function returns the global dispo quantity of a requisition line item.

Type: Function

Usage: R.20.01

Event: MTO Job

Default: See CIP source code

**STD\_JOB\_ID**

Description: This procedure is called by m_pck_mto.process_reqs and m_pck_req.check_req_aprv_preconditions.

Type: Procedure

Usage: R.20.01, R.30.01

Event: MTO Job

Default: m_pck_account_codes.assign_ac_to_rli(p_r_id)
Custom Packages

STD_JOB_INIT
Description: This procedure initializes an MTO job. The procedure is called by m_pck_mto.init_job.
Type: Procedure
Usage: R.20.01
Event: MTO Job
Default: m_pck_account_codes.std_job_init

STD_PROC_CHANNEL
Description: This procedure is called by m_pck_mto.process_reqs.
Type: Procedure
Usage: R.20.01, R.30.01
Event: MTO Job
Default: NULL

STD.Reverse.REL
Description: This procedure is called by m_pck_rel_to_proc.rev_rel_to_proc.
Type: Procedure
Usage: R.40.11
Event: Update requisition status
Default: NULL
Custom Packages

**STD_ROS_DATE**

Description: This procedure is called by m_pck_mto.process_reqs and m_pck_req.check_req_aprv Preconditions.

Type: Procedure

Usage: R.20.01, R.30.01

Event: MTO Job

Default: NULL

**SUM_INQUIRY**

Description: This procedure returns a value for the inquiry quantity, instead of m_pck_ident_sums.sum_inquiry.

Type: Procedure

Usage: C.40.01

Event: Query ident

Default: 0

**TECH_EVAL**

Description: This procedure is called by m_pck_mto.process_reqs.

Type: Procedure

Usage: R.20.01

Event: MTO Job

Default: NULL
**UPDATE_INQ_REQ**

Description: This procedure is called by m_pck_mto.update_inq_req. This procedure is executed when an inquiry only requisition is revised (that is, the original supplement 0 is updated instead of creating a new supplement).

- `p_temp_new_suppl_r_id` is the internal primary key of the supplement 1 requisition that is the source of the update but exists only temporarily, which means that after this CIP has been called, `m_pck_mto.update_inq_req` will delete this temporary supplement and unapprove ("open") supplement 0 again.
- `p_original_inq_req_r_id` is the internal PK of the revised (that is, suppl 0) inquiry requisition.

When this CIP is called, all updates to supplement 0 have already been made. Possible applications of this CIP are, for example:

- calling other API routines to external interfaces that need to be informed about this internal event
- reconciling any attachment data from supplement 1 to supplement 0 that is not done automatically by standard MTO functionality.

**WEIGHT**

Description: This function returns the weight of an ident. The function is called by `m_pck_req.calc_weight`.

**Type:** Function

**Usage:** AS.10.02, C.30.01, C.30.11, P.50.07, R.20.01, R.30.01

**Event:** Query ident, MTO Job

**Default:** See CIP source code
WEIGHT_UNIT
Description: This function returns the weight unit of an ident. The function is called by m_pck_req.calc_weight.
Type: Function
Usage: AS.10.02, C.50.31, E.10.11, P.50.07, R.20.01, R.30.01
Event: Query ident, MTO Job
Default: Unit code assigned to the weight attribute

M_PCK_S50R10_CUSTOM
This package is the report S.50.R.10 custom package. You can implement CIP PL/SQL functions to retrieve the information you need. Furthermore, S50R10.RDF can be edited in order to change prompts, and so on.

GET_CORRAL
Description: This function returns the corrosion allowance. The input parameter is the spec_header_id. The datatype of the return value is varchar2.
Type: Function
Usage: S.50.R.10
Event: Run report
Default: 'm_pck_s50R10_custom.get_corral'

GET_FACING
Description: This function returns the facing. The input parameter is the spec_header_id. The datatype of the return value is varchar2.
Type: Function
Usage: S.50.R.10
Event: Run report
Custom Packages

Default: 'm_pck_s50R10_custom.get_facing'

GET_IDENT_SHORT
Description: This function returns the ident short description by means of the CIP function M_PCK_CUSTOM.IDENT_DESC.
Type: Function
Usage: S.50.R.10
Event: Run report
Default: m_pck_std_custom.ident_desc (P_IDENT, P_NLS_ID)

GET_MATL
Description: This function returns the material. The input parameter is the spec_header_id. The datatype of the return value is varchar2.
Type: Function
Usage: S.50.R.10
Event: Run report
Default: 'm_pck_s50R10_custom.get_matl'

GET_MAX_PRESS
Description: This function returns the maximum pressure, that is the maximum value of column PRESSURE in table M_SPEC_LIMITS. The input parameter is the spec_header_id. The datatype of the return value is number.
Type: Function
Usage: S.50.R.10
Event: Run report
Default: max(pressure) from m_spec_limits
**GET_MAX_TEMP**

Description: This function returns the maximum temperature, that is the maximum value of column TEMP in table M_SPEC_LIMTS. The input parameter is the spec_header_id. The datatype of the return value is number.

Type: Function

Usage: S.50.R.10

Event: Run report

Default: max(temp) from m_spec_limits

**GET_PLANT_LOC**

Description: This function returns the project location from project default ZX_PROJLOC.

Type: Function

Usage: S.50.R.10

Event: Run report

Default: m_pck_ppd_defaults.get_value('ZX_PROJLOC')

**GET_PROJ_NR**

Description: This function returns the project number from project default ZX_PROJ_NR.

Type: Function

Usage: S.50.R.10

Event: Run report

Default: m_pck_ppd_defaults.get_value('ZX_PROJ_NR')
Custom Packages

GET_RATING
Description: This function returns the rating. The input parameter is the spec_header_id. The datattype of the return value is varchar2.
Type: Function
Usage: S.50.R.10
Event: Run report
Default: 'm_pck_s50R10_custom.get_rating'

GET_SERVICE
Description: This function returns the service. The input parameter is the spec_header_id. The datattype of the return value is varchar2.
Type: Function
Usage: S.50.R.10
Event: Run report
Default: 'm_pck_s50R10_custom.get_service'

M_PCK_S50R11_CUSTOM
This package is the report S.50.R.11 custom package. You can implement CIP PL/SQL functions to retrieve the information you need. Furthermore, S50R11.RDF can be edited in order to change prompts, and so on.

GET_BRANCH_TABLE
Description: This function returns the name of the branch table that is assigned to the specification via m_spec_header_geometrics.
Type: Function
Usage: S.50.R.11
Event: Run report
Custom Packages

Default: 
M_GEOM_BRANCHES.GB_CODE = M_SPEC_HEADER_GEOMETRICS.TABLE_NAME where KIND_OF='BRANCHES'

GET_COMMODITY

Description: This function returns the commodity code. The input parameter is the commodity_id. The datatype of the return value is varchar2.

Type: Function

Usage: S.50.R.11

Event: Run report

Default: m_commodity_codes.commodity_code

GET_COMMODITY_DESC

Description: This function returns the NLS dependent commodity code description. The input parameter is the commodity_id and the nls_id. The datatype of the return value is varchar2.

Type: Function

Usage: S.50.R.11

Event: Run report

Default: m_commodity_code_nls.short_desc

GET_CORRAL

Description: This function returns the corrosion allowance. The input parameter is the spec_header_id. The datatype of the return value is varchar2.

Type: Function

Usage: S.50.R.11

Event: Run report
Custom Packages

Default: 'm_pck_s50R11_custom.get_corral'

**GET_CTRLL_SPEC**

Description: This function returns the ctrll (control level). The input parameter is the spec_header_id. The datatype of the return value is varchar2.

Type: Function
Usage: S.50.R.11
Event: Run report
Default: 'm_pck_s50r11_custom.GET_CTRLL_SPEC'

**GET_DESIGN_STD**

Description: This function returns the design standard. The input parameter is the spec_header_id. The datatype of the return value is varchar2.

Type: Function
Usage: S.50.R.11
Event: Run report
Default: 'm_pck_s50r11_custom.GET_DESIGN_STD'

**GET_DETAIL_CONSTRUC**

Description: This function returns the detail construction. The input parameter is the spec_header_id. The datatype of the return value is varchar2.

Type: Function
Usage: S.50.R.11
Event: Run report
Default: 'm_pck_s50r11_custom.GET_DETAIL_CONSTRUC'
**GET_DETAIL_FINISH**

Description: This function returns the detail finish. The input parameter is the spec_header_id. The datatype of the return value is varchar2.

Type: Function

Usage: S.50.R.11

Event: Run report

Default: 'm_pck_s50r11_custom.GET_DETAIL_FINISH'

---

**GET_DETAIL_MATERIAL**

Description: This function returns the detail material. The input parameter is the spec_header_id. The datatype of the return value is varchar2.

Type: Function

Usage: S.50.R.11

Event: Run report

Default: 'm_pck_s50r11_custom.get_detail_material'

---

**GET_DETAIL_RATING**

Description: This function returns the detail rating. The input parameter is the spec_header_id. The datatype of the return value is varchar2.

Type: Function

Usage: S.50.R.11

Event: Run report

Default: 'm_pck_s50r11_custom.GET_DETAIL_RATING'
Custom Packages

GET_DETAIL_SCH
Description: This function returns the detail schedule. The input parameter is the spec_header_id. The datatype of the return value is varchar2.
Type: Function
Usage: S.50.R.11
Event: Run report
Default: 'm_pck_s50r11_custom.get_detail_SCH'

GET_END1
Description: This function returns the end condition 1. The input parameter is the spec_header_id. The datatype of the return value is varchar2.
Type: Function
Usage: S.50.R.11
Event: Run report
Default: 'm_pck_s50r11_custom.GET_END1'

GET_END2
Description: This function returns the end condition 2. The input parameter is the spec_header_id. The datatype of the return value is varchar2.
Type: Function
Usage: S.50.R.11
Event: Run report
Default: 'm_pck_s50r11_custom.GET_END2'
GET_FACING
Description: This function returns the facing. The input parameter is the spec_header_id. The datatype of the return value is varchar2.
Type: Function
Usage: S.50.R.11
Event: Run report
Default: 'm_pck_s50R11_custom.get_facing'

GET.HEAT_SPEC
Description: This function returns the heat specification. The input parameter is the spec_header_id. The datatype of the return value is varchar2.
Type: Function
Usage: S.50.R.11
Event: Run report
Default: 'm_pck_s50r11_custom.GET_HEAT_SPEC'

GET_MATL
Description: This function returns the material. The input parameter is the spec_header_id. The datatype of the return value is varchar2.
Type: Function
Usage: S.50.R.11
Event: Run report
Default: 'm_pck_s50R11_custom.get_matl'
Custom Packages

GET_MAX_PRESS

Description: This function returns the maximum pressure, that is, the maximum value of column PRESSURE in table M_SPEC_LIMITS. The input parameter is the spec_header_id. The datatype of the return value is number.

Type: Function
Usage: S.50.R.11
Event: Run report
Default: max(pressure) from m_spec_limits

GET_MAX_TEMP

Description: This function returns the maximum temperature, that is, the maximum value of column TEMP in table M_SPEC_LIMITS. The input parameter is the spec_header_id. The datatype of the return value is number.

Type: Function
Usage: S.50.R.11
Event: Run report
Default: max(temp) from m_spec_limits

GET_MIN_PRESS

Description: This function returns the minimum pressure, that is, the minimum value of column PRESSURE in table M_SPEC_LIMITS. The input parameter is the spec_header_id. The datatype of the return value is number.

Type: Function
Usage: S.50.R.11
Event: Run report
Default: min(pressure) from m_spec_limits
Custom Packages

GET_MIN_TEMP
Description: This function returns the minimum temperature, that is, the minimum value of column TEMP in table M_SPEC_LIIMITS. The input parameter is the spec_header_id. The datatype of the return value is number.
Type: Function
Usage: S.50.R.11
Event: Run report
Default: min(temp) from m_spec_limits

GET_PLANT_LOC
Description: This function returns the project location from project default ZX_PROJLOC.
Type: Function
Usage: S.50.R.11
Event: Run report
Default: m_pck_ppd_defaults.get_value('ZX_PROJLOC')

GET_PROJ_NR
Description: This function returns the project number from project default ZX_PROJ_NR.
Type: Function
Usage: S.50.R.11
Event: Run report
Default: m_pck_ppd_defaults.get_value('ZX_PROJ_NR')
Custom Packages

GET_RATING
Description: This function returns the rating. The input parameter is the spec_header_id. The datatype of the return value is varchar2.
Type: Function
Usage: S.50.R.11
Event: Run report
Default: 'm_pck_s50R11_custom.get_rating'

GET_SCHEDULE
Description: This function returns the schedule. The input parameter is the spec_header_id. The datatype of the return value is varchar2.
Type: Function
Usage: S.50.R.11
Event: Run report
Default: 'GET_SCHEDULE'

GET_SERVICE
Description: This function returns the service. The input parameter is the spec_header_id. The datatype of the return value is varchar2.
Type: Function
Usage: S.50.R.11
Event: Run report
Default: 'm_pck_s50R11_custom.get_service'
**GET_SIZES**

Description: This function returns sizes. The input parameter is the spec_header_id. The datatype of the return value is number.

Type: Function

Usage: S.50.R.11

Event: Run report

Default: 1234

**GET_THK**

Description: This function returns wall thickness. The input parameter is the spec_header_id. The datatype of the return value is number.

Type: Function

Usage: S.50.R.11

Event: Run report

Default: 5678

**GET_WELDING_SPEC**

Description: This function returns the welding specification. The input parameter is the spec_header_id. The datatype of the return value is varchar2.

Type: Function

Usage: S.50.R.11

Event: Run report

Default: 'm_pck_s50r11_custom.GET_WELDING_SPEC'
M_PCK_SAP_CUSTOM

This package contains SAP CIP functions.

BUILD_SAP_ART_NR

Description: CIP to build the SAP article number. This special function builds the SAP article number from the ident code. It is not possible to use this function as a normal SAP CIP function, because of the number and type of the input parameters.

Type: Function
Usage: SAP Interface
Event: Start button on SAP.20.01/SAP.50.01 (if set in SAP definitions)
Default: 'M'||substr ( m_ids.ident_code, 1, 10)

CHECK_TRANSFER

Description: This function is for SAP requisition transfer. The function returns TRUE if the ident was transferred to SAP. Otherwise, it returns FALSE.

Type: Function
Usage: SAP Interface
Event: Start button on SAP.20.01/SAP.50.01 (if set in SAP definitions)
Default: TRUE, FALSE

FETCH_ECLASS

Description: Fetch_EClass. This function can be used for segment 'E1KSSD', field 'CLASS'

> fetch SAP MATERIAL CLASS on SAP.10.07

Type: Function
Usage: SAP Interface
Event: Start button on SAP.20.01/SAP.50.01 (if set in SAP definitions)
FETCH_MATNR

Description: This function fetches the material number.
Type: Function
Usage: SAP Interface
Event: Start button on SAP.20.01/SAP.50.01 (if set in SAP definitions)
Default: m_idents.ids_code if m_pck_sap.g_company_id is NULL else <PROJ_ID>.m_idents.ident_code

FETCH_PLANT

Description: This function fetches the plant code.
Type: Function
Usage: SAP Interface
Event: Start button on SAP.20.01/SAP.50.01 (if set in SAP definitions)
Default: NULL

GET_CHANGED_IDENTS

Description: This function identifies idents that were changed since the last transfer to SAP.
Type: Function
Usage: SAP Interface
Event: Start button on SAP.20.01 (only for the Changed Idents transfer mode)
Default: m_pck_sap.Changed_Idents_Tab
Custom Packages

GET_DESC

Description: This function returns the NLS dependent commodity code short description.
Type: Function
Usage: SAP Interface
Event: Start button on SAP.20.01/SAP.50.01 (if set in SAP definitions)
Default: mvp_commodity_code_nls.short_desc

GET_GEWICHT

Description: This function returns the ident weight.
Type: Function
Usage: SAP Interface
Event: Start button on SAP.20.01/SAP.50.01 (if set in SAP definitions)
Default: '2211,5535'

GET_IDENT_FROM_RLI

Description: This function is for SAP requisition transfer. The function returns the ident code of the requisition line item ident.
Type: Function
Usage: SAP Interface
Event: Start button on SAP.50.01 (if set in SAP definitions)
Default: m_idents.ident_code where ident = m_req_line_items.ident

226 SmartPlant Materials Customer-Installed Procedures (CIPs)
Custom Packages

GET_INTERFACE_ID
Description: This function returns the current interface ID.
Type: Function
Usage: SAP Interface
Event: Start button on SAP.20.01/SAP.50.01 (if set in SAP definitions)
Default: p_interface_id

GET_LONG
Description: CIP function for testing.
Type: Function
Usage: SAP Interface
Event: Start button on SAP.20.01/SAP.50.01 (if set in SAP definitions)
Default: test string

GET_LONGDESC
Description: This function returns the NLS dependent commodity code description.
Type: Function
Usage: SAP Interface
Event: Start button on SAP.20.01/SAP.50.01 (if set in SAP definitions)
Default: mvp_commodity_code_nls.description
Custom Packages

GET_LONG_DESCRIPTION
Description: CIP function for testing.
Type: Function
Usage: SAP Interface
Event: Start button on SAP.20.01/SAP.50.01 (if set in SAP definitions)
Default: test string

GET_MEINS_NENNER
Description: Internal function for weight handling.
Type: Function
Usage: SAP Interface
Event: Start button on SAP.20.01/SAP.50.01 (if set in SAP definitions)
Default: ''

GET_MEINS_ZAEHLER
Description: Internal function for weight handling.
Type: Function
Usage: SAP Interface
Event: Start button in SAP.20.01/SAP.50.01 (if set in SAP definitions)
Default: ''
**GET_REQ_DESCRIPTION**

Description: This function is for SAP requisition transfer. This function returns the NLS dependent requisition description.

Type: Function

Usage: SAP Interface

Event: **Start** button on SAP.50.01 (if set in SAP definitions)

Default: `m_req_nls.description`

**GET_REQ_NAME**

Description: This function is for SAP requisition transfer. This function returns the requisition name.

Type: Function

Usage: SAP Interface

Event: **Start** button on SAP.50.01 (if set in SAP definitions)

Default: `m_reqs.r_code`

**GET_RLI_IDENTDESC**

Description: This function is for SAP requisition transfer. The function returns the NLS dependent description of the ident on the requisition line item.

Type: Function

Usage: SAP Interface

Event: **Start** button on SAP.50.01 (if set in SAP definitions)

Default: `m_pck_std_custom.ident_desc(ident,nls_id)` where `ident=m_req_line_items.ident`
**GET_RLI_ROSDATE**

Description: This function is for SAP requisition transfer. The function returns the ROS (required on site) date of the requisition line item.

Type: Function

Usage: SAP Interface

Event: Start button on SAP.50.01 (if set in SAP definitions)

Default: m_req_line_items.estimated_ros_date

---

**GET_RLI_UNIT**

Description: This function is for SAP requisition transfer. The function returns the quantity unit of the requisition line item.

Type: Function

Usage: SAP Interface

Event: Start button on SAP.50.01 (if set in SAP definitions)

Default: m_units.unit_code where unit_id=m_req_line_items.qty_unit_id

---

**GET_RLI_WEIGHT_UNIT**

Description: This function is for SAP requisition transfer. The function returns the weight unit of the requisition line item.

Type: Function

Usage: SAP Interface

Event: Start button on SAP.50.01 (if set in SAP definitions)

Default: m_units.unit_code where unit_id=m_req_line_items.unit_id


GET_SHORTDESC

Description: This function returns the NLS dependent ident short description.
Type: Function
Usage: SAP Interface
Event: Start button on SAP.20.01/SAP.50.01 (if set in SAP definitions)
Default: mvp_ident_nls.short_desc

GET_SYSDATE

Description: CIP function for testing.
Type: Function
Usage: SAP Interface
Event: Start button on SAP.20.01/SAP.50.01 (if set in SAP definitions)
Default: to_char(sysdate, 'YYYYMMDD')

GET_TAG_TEXT

Description: This function returns a concatenation of ident, commodity code, and ident description.
Type: Function
Usage: SAP Interface
Event: Start button on SAP.20.01/SAP.50.01 (if set in SAP definitions)
Default: substr(i.ident||cc.commodity_code||m_pck_std_custom.ident_desc(i.ident, p_nls_id), 1, 4000) from mvp_idents i, mvp_commodity_codes cc
IDENT_CHANGE

Description: This function checks if an ident was changed since the last transfer to SAP, and if the last transfer to SAP was faulty. In these cases, Y is returned, otherwise N.

Type: Function

Usage: SAP Interface

Event: Start button on SAP.20.01/SAP.50.01 (if set in SAP definitions)

Default:

M_PCK_SECURITY_CUSTOM

Supplies a CIP function to validate password complexity.

CHECK_COMPLEX_PASSWORD

Description: This function checks a user password. The function returns TRUE if the password contains characters, numbers, and special characters.

Type: Function

Usage: A.60.01.01, A.60.01.03, Logon, Logon_Link

Event: Create or change user password

Default: TRUE if the password contains characters, numbers, and special characters; otherwise, FALSE.

GET_DISABLED_FK

Description: This function delivers the count of disabled foreign keys.

Type: Function

Usage: Logon

Event: Open logon screen
Default: select count(*) from dba_constraints where owner = 'M_SYS' and constraint_type = 'R' and status = 'DISABLED';

GET_LAST_ANALYZED

Description: This function delivers the MIN last_analyzed date from user_tables.

Type: Function

Usage: Logon

Event: Open logon screen

Default: min(last_analyzed) from user_tables

M_PCK_SE_CUSTOM

Contains all customer functions for downloading data to an ASCII file (C.10.15 Site Export Jobs).

FIELDS_PIPE_SPEC

Description: Customer-specific example.

This function can be used to write additional information to the ASCII file for the table section 'Material Specification', depending on the ident. This information is written at the end of the row. The row contains the following fixed fields:

Ident Code, Ident Short Desc, TAG Number, Commodity Code, Size 1, Size 2, Size 3, Size 4, Size 5, Part Code, Short Code, Unit Set

Type: Function

Usage: C.10.15

Event: Click the Start button.

Default: See sample source code.
**MAIN_MATERIAL**

Description: Customer-specific example.

This function returns the main material, depending on the selected commodity code. The main material is set in the fourth commodity code digit, with the following meaning:

- A = Carbon Steel
- B = Low Alloy Steel
- C = Stainless Steel
- D = Iron
- E = Non-Ferrous
- F = Non-Metallic

In the case of part code 411, the value for the main material is ‘*’

Type: Function

Usage: C.10.15

Event: Click the **Start** button.

Default: See sample source code.

---

**PROMPT_PIPE_SPEC**

Description: Customer-specific example.

This function can be used to define the prompts of the additional information in the ASCII file for the the table section 'Material Specification'. These prompts are written to the end of the row. The row contains the following fixed prompts:

- Ident Code, Ident Short Desc, TAG Number, Commodity Code, Size 1, Size 2, Size 3, Size 4, Size 5, Part Code, Short Code, Unit Set

Type: Function

Usage: C.10.15

Event: Click the **Start** button.

Default: See sample source code.
**TRAFFIC**

Description: CIP function for traffic dates download.

Type: Function

Usage: C.10.15

Event: Click the **Start** button.

Default: NULL

---

**M_PCK_SITE_CUSTOM**

This package contains all customizable procedures and functions for Site.

---

**BEFORE_POST_MIR**

Description: Used by MIR screens. The procedure is executed by m_pck_inv_items.issue_inv_item (Post button).

Type: Procedure


Event: Click the **Post** button.

Default: NULL

---

**CALC_SELL_PRICES**

Description: Called from C.20.04 Prepare MRR by Release Notes after the Populate Selling Price button is clicked. The procedure populates the selling price to the release note and package items.

Type: Procedure

Usage: C.20.04

Event: Click the **Populate Selling Price** button.

Default: NULL
CHECK_BEFORE_POST_MRR

Description: Used by C.20.01-.05 MRR screens. The function is called by the screen before an MRR is posted. You can use this CIP to implement any checks or actions that must be performed before the MRR is posted.

Type: Function

Usage: C.20.01, C.20.01.02, C.20.03, C.20.04, C.20.05, C.20.07

Event: Click the Post button.

Default: TRUE

CHECK_BEFORE_POST_MTR

Description: Used by C.20.15 MTR screen. The function is called by the screen before an MTR is posted and can be used to perform validations of the material transfer reports before the MTR is posted.

Type: Function

Usage: C.20.15

Event: Click the Post button.

Default: TRUE

CHECK_BEFORE_UNPOST

Description: Used by C.20.01-.05 MRR screens. The function is called by a database table trigger before an MRR is unposted.

Type: Function

Usage: C.20.01, C.20.01.02, C.20.03, C.20.04, C.20.05

Event: Click the Unpost button.

Default: TRUE
CHECK_FIELDS

Description: Used by C.20.01-05 MRR screens. The function is called by m_pck_receive.receive_inv_item (Post button) to check if some necessary fields on C.20.08 are filled (TRUE) or not (FALSE).

Type: Function
Usage: C.20.08
Event: Click the Post button.
Default: TRUE

CHECK_HEAT_NUMBER_ISSUE_QTY

Description: Used by the C.20.21-23 MIR screens. The function is executed by m_pck_inv_items.issue_inv_item (Post button) to check if the sum of the heat number quantities is equal to the issue qty (TRUE) or not (FALSE).

Type: Function
Usage: C.20.25, C.20.25.02
Event: Click the Post button.
Default: TRUE

CHECK_HEAT_NUMBER_RECEIVED

Description: Used by C.20.21-23 MIR screens. The function is executed by m_pck_inv_items.issue_inv_item (Post button) to check if the issued heat numbers have been received (TRUE) or not (FALSE).

Type: Function
Usage: C.20.25, C.20.25.02
Event: Click the Post button.
Default: TRUE
### CHECK_MRR_RECV_QTY

**Description:** Used by C.20.01-05 MRR screens. The function is called by `m_pck_receive.receive_inv_item (Post button)` to check if the sum of the heat number quantities is equal to the received qty (TRUE) or not (FALSE).

**Type:** Function

**Usage:** C.20.08

**Event:** Click the **Post** button.

**Default:** TRUE

### CHECK_OSDS

**Description:** Used by C.20.01-05 MRR screens. The function is called by `m_pck_receive.receive_inv_item (Post button)` to check the OSD quantities.

**Type:** Function

**Usage:** C.20.01, C.20.01.02, C.20.03, C.20.04, C.20.05, C.20.07

**Event:** Click the **Post** button.

**Default:** See CIP source code

### CHECK_VENDOR_CODE

**Description:** Used by C.20.21-23 MIR screens. The function is called by `m_pck_inv_items.issue_inv_item (Post button)` to check if the vendor code is correct (TRUE) or not (FALSE).

**Type:** Function

**Usage:** C.20.25, C.20.25.02

**Event:** Click the **Post** button.

**Default:** TRUE
ERP_TRANSFER

Description: Called from C.20.04 Prepare MRR by Release Notes when the ERP Transfer button is clicked.

Type: Procedure

Usage: C.20.04

Event: Click the ERP Transfer button.

Default: SET m_matl_recv_rpts.erp_set_ind = 'Y'

ON_SITE_INSPECTION_STATUS_CHANGE

Description: This procedure is executed whenever the site inspection status is changed on the C.50.11 Site Inspection screen.

Type: Procedure

Usage: C.50.11

Event: Change site inspection status

Default: NULL

POST_DELETE_MRR

Description: This function is called after an MRR has been deleted in table m_matl_recv_rpts. If this function runs successfully, TRUE is returned. If there is an error, a message is raised.

Type: Function

Usage: Site

Event: Delete MRR

Default: TRUE
POST_MIR_CREATION

Description: This procedure is executed after a MIR (material issue report) is created. This procedure can be used, for example, to automatically assign values to the MIR, such as an approval sequence.

Type: Procedure

Usage: C.20.21
C.20.22.01
C.20.22.02
C.20.22.04
C.20.23.01
C.20.23.02
C.20.21.11

Event: Create MIR

Default: NULL

SITE_INSPECTION

Description: Used by C.50.11 Site Inspection. This procedure is executed when the Execute CIP button is clicked.

Type: Procedure

Usage: C.50.11

Event: Click the Execute CIP button.

Default: NULL

M_PCK_SM_CUSTOM

This package contains all procedures and functions that can be customized, in the area of subcontract management.
POST_APPROVE_PERIOD_PROGRESS
Description: This CIP function is called after a period progress is approved on SM.20.11 Subcontract Administration, but before the approval is stored.
If this function runs successfully, TRUE is returned. If there is an error, a message is raised, and the approval is reversed.
Type: Function
Usage: SM.20.11
Event: Period progress approval
Default: TRUE

POST_DELETE_PERIOD_PROGRESS
Description: This CIP function is called after a period progress is deleted on SM.20.11 Subcontract Administration.
If this function runs successfully, TRUE is returned. If there is an error, a message is raised.
Type: Function
Usage: SM.20.11
Event: Delete period progress
Default: TRUE

POST_SUBMIT_PERIOD_PROGRESS
Description: This CIP function is called after the approval of a period progress is submitted on SM.20.11 Subcontract Administration, but before the submittal is stored.
If this function runs successfully, TRUE is returned. If there is an error, a message is raised.
Type: Function
Usage: SM.20.11
Event: Submit period progress
Custom Packages

Default: TRUE

POST_UNAPPROVE_PERIOD_PROGRESS

Description: This CIP function is called after the approval of a period progress is reversed on **SM.20.11 Subcontract Administration**, but before the change is saved. If this function runs successfully, TRUE is returned. If there is an error, a message is raised.

Type: Function
Usage: SM.20.11
Event: Reverse approval of period progress
Default: TRUE

M_PCK_SP3DEXP_CUSTOM

This package is for CIP procedures related to Intergraph Smart™ 3D export.

PUBLISH

Description: This procedure is executed after the Smart 3D job has finished successfully.

Type: Procedure
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: update m_sp3d_exp_dst_dta_si set ej_id = p_ej_id where ed_id in (select ed_id from m_sp3d_exp_jobs where ej_id=p_ej_id)
M_PCK_SP3D_CUSTOM

This custom package writes data into Smart 3D interface tables.

ADD_CPN_STRING

Description: This procedure adds one token of the catalog part number.
Type: Procedure
Usage: SP3D.20.01
Event: Smart 3D Export Job
Default: See CIP source code

ADD_ZERO

Description: This function adds .0mm.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: IF instr(P_SCH,'.') = 0 then P_SCH||'.0mm' else P_SCH||'mm'

ANALYZE_LINE

Description: This function is needed during BLL (bulkload log) analysis to categorize the log file contents.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code
Custom Packages

**BUILD_CATALOG_PART_NUMBER**

Description: This function is needed by \_pck\_sp3d\_custom.set\_catalog\_part\_number. It builds the string forming the catalog part number.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code

**CGR_ATTR_VALUE**

Description: This function returns an attribute value from geometrics based on matching of ident input columns with no unit conversion. For unit conversion, use the version in sdb_sys.sdb_pck_sp3d_custom.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code

**CONVERT_PHY_DIM**

Description: This function is used to convert dimensions of physical attributes.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code
CONVERT_SIZE

Description: This function is used to convert size dimensions.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code

CONVERT_SIZE_FROM_ASM

Description: This function converts value like 25.4mm to 1in. The target unit code is set by g_sp3d_uom.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: convert_size (TRIM (SUBSTR (p_size,1,LENGTH (p_size) - 2)),v_source_uom))

DELETE_DUP_VALVE_TAG_IDENTS

Description: This procedure deletes duplicate entries produced by valve tag idents from the export job interface tables.
Type: Procedure
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code
**FETCH_CPN_CUR**

Description: This function fetches from a dynamic cursor, if it is needed for building the catalog part number.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code

**FILL GENERIC BOLTED**

Description: This procedure fills the generic bolted data.

Type: Procedure

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code

**FILL GENERIC FEMALE**

Description: This procedure fills the generic female data.

Type: Procedure

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code
FILL.PLAIN GENERIC

Description: This procedure fills the plain generic data.
Type: Procedure
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code

FILL_SPEC_ITEM_IDENT_PG_TEMP

Description: This procedure fills the mv_spec_item_ident_pg_temp table.
Type: Procedure
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code

FILL_SPEC_RULE_SHEETS

Description: This procedure generates the spec rule sheet data for a given spec. Currently, it covers the nut selection filter, bolt selection filter, gasket selection filter, and washer selection filter. This function can be extended easily to create data for other spec rule sheets such as bolt extension or minimum pipe length.
Type: Procedure
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code
Custom Packages

**FILL_VALVE_OP_CTRL_DTA**

Description: This procedure fills the valve operator control data.

Type: Procedure

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code

**FILTER_IDENTS_USING_ATTR_VALS**

Description: This procedure filters idents based on attribute values on SP3D 10.04.

Type: Procedure

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code

**FIND_IN_CC_ENDPREP**

Description: This function returns the end preparation 1 value if null.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code
**FND_IN_CC_2ND_THEN_1ST**

**Description:** This function retrieves the value from commodity code detail for the second connect point. If the value is not there, it takes the value from the first connect point.

**Type:** Function

**Usage:** SP3D 20.01

**Event:** Smart 3D Export Job

**Default:** See CIP source code

---

**FND_IN_CC_OR_1004**

**Description:** This function retrieves a value from the commodity code detail or from the mapping on SP3D 10.04. If the value is not found in one source, it is looked up in the other. By setting `p_priority` to either CC or 1004, you decide which one should be tried first.

**Type:** Function

**Usage:** SP3D 20.01

**Event:** Smart 3D Export Job

**Default:** See CIP source code

---

**GET_ATTR_VAL_AND_UNIT**

**Description:** This function returns the value and unit of a physical attribute.

**Type:** Function

**Usage:** SP3D 20.01

**Event:** Smart 3D Export Job

**Default:** See CIP source code
Custom Packages

GET_BCGNW_TYPE
Description: This function is needed by m_pck_lookup.get_commodity_type. To address the correct columns, it is necessary to know if it is bolt/gasket/nut/washer or commodity.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: B for bolts, G for gaskets, N for nuts, W for washers; otherwise, C

GET_CGR_ATTR_VALUE
Description: This function returns the physical attribute value of a geometric detail.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code

GET_COMP_WEIGHT
Description: This function returns the weight of a component.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code
**GET_CUSTOM_SHORT_CODE**

Description: This function modifies the short code for Smart 3D.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code

**GET_DETAIL_ID**

Description: This function returns an object parameter detail.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code

**GET_EXP_TXT_FILE_NAME**

Description: This function returns the string to be used as the operating system name for exported text files. The function is intended to be able to add project, job name, and run number if required by the customer.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code
GET_EXP_WBK_NAME
Description: This function returns the string to be used as the file name for exported sheets. The function is intended to be able to add project, job name, and run number if required by the customer. Workbook type CDLST/OTHER or at least some difference in name must be given. Also, modify get_wbk_type if changed. The xls file type is appended by the software, because it is needed by other functions.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code

GET_EXP_ZIP_NAME
Description: This function returns the string to be used as the zip file name for exported text files. The function is intended to be able to add project, job name, and run number if required by the customer. The zip file type is appended by the software, because it is needed by other functions.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: _CDLST when category='CODE_LIST', else _OTHER

GET_GEN_ATTR_VALUE
Description: This function returns an attribute value from the geometric details.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code
**GET_GEN_ENDPREP**

Description: This function returns the end preparation value.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: `m_table_details.attr_num1 where table_name='P_END_PREP'`

---

**GET_GEN_ENDSTD**

Description: This function returns the dimensional standard value.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: `m_table_details.attr_num2 where table_name='P_DIM_STD'`

---

**GET_GEN_RATING**

Description: This function returns the rating value.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: `m_table_details.attr_char1 where table_name='P_RATING'`
Custom Packages

GET_GN_CONVERTED
Description: This function handles the English-Metric conversion.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code

GET_IDENT_SIZE_SCH
Description: This function controls whether the procedure will try to use 1SCH for 2SCH, 1NS for 2NS.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code

GET_IDENT_VALUE
Description: This function returns an ident value.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code
**GET_IDENT_VALUE_PM**

Description: This function returns an ident value with appended /m (per meter).

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: get_ident_value(P_IDENT,P_INTERFACE_code,p_translation)||'\m'

**GET_LOG_HEADER_PARM**

Description: This function is needed during BLL (bulkload log) analysis to fetch initial parameters from the log file contents, like the bulkload mode, catalog server name, and so on.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: LTRIM(RTRIM(m_pck_string_utl.rightpart(p_linebuf,':')))

**GET_MAPPED_PARTCLASS**

Description: This function checks which idents belong to which part class.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code
**GET_MASTER_CDLST_NUM**

Description: This function is a simple wrapper function of the Smart 3D interface API `get_master_cdlst_number`.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code

---

**GET_MATERIAL_DESCRIPTION**

Description: This function returns the description for a commodity code and/or ident. There are six samples that can be used by themselves or combined. You can also create your own material description. This function is called, for example, to fill in the PipingCommodityMatiControlData sheet.

Type: Function

Usage: SP3D 10.12

Event:

Default: See CIP source code

---

**GET_MATERIAL_LABEL_DESC**

Description: This function returns a Smart 3D label that is used in the piping material control data, description attribute. The label value is derived from the commodity type of this commodity code. The mapping between the label and commodity code is done on the template codelist sheet CommodityType.

Type: Function

Usage: SP3D 10.12

Event:

Default: See CIP source code
GET_MATERIAL_SHORT_DESC

Description: This function returns the short description for a commodity code and/or ident. There are six samples that can be used by themselves or combined. You can also create your own material short description. This function is called, for example, to fill in the PipingCommodityMatlControlData sheet.

Type: Function
Usage: SP3D 10.12
Event:
Default: See CIP source code

GET_NLS_CNT

Description: This function is intended to set the sheet category to GENERAL_LIST.

Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: SELECT COUNT(nls_id) FROM m_note_nls

GET_NPS_CONVERTED

Description: This function handles the NPS unit conversion.

Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code
Custom Packages

GET_NPS_SCH
Description: This function determines the object parameter details for the commodity codes and returns the requisite input_1 – input_4 value for the PipingCommodityProcurementData sheet.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code

GET_PORT
Description:
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code

GET_PORT_3
Description:
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code
**GET_SHT_NAME**

Description: This function is needed during BLL (bulkload log) analysis to fetch the sheet name from the log file contents. This can be done on the sheet start and sheet end.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code

**GET_SI_MULTISIZEOPTION**

Description: This function is intended to get the MultiSizeOption from specification items.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code

**GET_SI_SCHEDULE**

Description:

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code
GET_SI_SIZE_RANGE

Description:
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code

GET_SI_SIZE_RANGE1

Description: This function returns the size from/to for the first spec item size range. The function checks if the short code is for a half coupling. If yes, the header size on input_1 is returned. The header size is retrieved from a spec header geometric that is assigned to the half coupling short code. All parts that are not a half coupling keep their original size.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code

GET_SI_SIZE_RANGE2

Description: This function returns the size from/to for the second spec item size range. The function checks if the short code is that for a half coupling. If yes, size range 1 is returned. All parts that are not a half coupling keep their original size.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code
**GET_SP3D_ATTR_VALUE**

Description: This function is used to find a Smart 3D value if retr_method is ATTR_STND_METHOD and the source is Default. This function does not cover all kinds of item mapping available in the Smart 3D interface. It can be extended later.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code

---

**GET_SP3D_UOM**

Description: This function returns the unit of measurement.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: g_sp3d_uom

---

**GET_SPEC_HEADER_ID**

Description: This function is used in the PIP report to return the specification header ID.

Type: Function

Usage: PIP report

Event: Run report

Default: see source code
Custom Packages

GET_STUB_LENGTH
Description: This function is used to get the length value for the Lap Joint Flange sheet.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code

GET_STUB_PATTERN
Description: This function returns the stub pattern.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: g_sp3d_uom

GET_STUBEND_CODE
Description: This function returns the stub end code.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code
**GET_TABLE_DETAIL**

Description: This function returns a table detail for a commodity code.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code

---

**GET_TD_DENSITY**

Description: This function returns the density value for a commodity code.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: $m_{\text{geom}}_{\text{details}}$.input_{1} \text{ where gn\_code='MATERIAL\_DENSITY'}$

---

**GET_TD_ENDPREP**

Description: This function returns the end preparation value for a commodity code.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: $m_{\text{table}}_{\text{details}}$.attr\_num1 \text{ where table\_name = 'P\_END\_PREP'}$

---
Custom Packages

GET_TD_GEOMSTD
Description: This function returns the dimensional standard value for a commodity code.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: m_table_details.attr_num1 where table_name = 'P_DIM_STD'

GET_TD_MATGRADE
Description: This function returns the material grade value for a commodity code.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: m_table_details.attr_num1

GET_TD_RATING
Description: This function returns the rating value for a commodity code.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: m_table_details.attr_char1 where table_name = 'P_RATING'
GET_TRANS_CC_GASKET
Description: This function returns the commodity code, appending the gasket thickness (ident input 2) value (separated by an underscore).
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: p_commodity_code ||"_"|| p_input_2

GET_TRANS_CC_GNW
Description: This function is for special handling for gaskets, nuts, and washers.
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code

GET_TRANS_CC_NUT
Description: This function returns the commodity code, appending the bolt diameter (ident input 1) and nut height (ident input 1) values (separated by an underscore).
Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: p_commodity_code ||"_"|| p_input_1||"_"|| p_input_2
**GET_TRANS_CC_WASHER**

Description: This function returns the commodity code, appending the bolt diameter (ident input 1) and washer thickness (ident input 2) values (separated by an underscore).

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: p_commodity_code ||"_"|| p_input_1||"_"|| p_input_2

**GET_UNIT_GSKTHK**

Description: This function returns the unit of the gasket thickness based on the commodity geometric table attribute unit type.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code

**GET_VALVE_CAT_NUM**

Description: This function reads the Smart 3D item mapping first. If no value is returned, the PDS item mapping is used. The PDS mapping value is returned by the original get_valve_cat_num function.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code
GET_VC_ENDPRP_VALUE

Description: This function retrieves the value for the end preparation from the pre-filled g_js_data_tab. This function is used during the verification of part class lines, to retrieve values for distinct properties.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code

GET_VC_ENDSTD_VALUE

Description: This function retrieves the value for the end standard from the pre-filled g_js_data_tab. This function is used during the verification of part class lines, to retrieve values for distinct properties.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code

GET_VC_NPD_VALUE

Description: This function retrieves the value for nominal size from the pre-filled g_js_data_tab. This function is used during the verification of part class lines, to retrieve values for distinct properties.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code
Custom Packages

**GET_VC_PRESSRAT_VALUE**

Description: This function retrieves the value for pressure/rating from the pre-filled g_js_data_tab. This function is used during the verification of part class lines, to retrieve values for distinct properties.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code

**GET_VC_SCH_VALUE**

Description: This function retrieves the value for schedule from the pre-filled g_js_data_tab. This function is used during the verification of part class lines, to retrieve values for distinct properties.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code

**GET_VC_UNIT_VALUE**

Description: This function retrieves the value for the unit of measure from the pre-filled g_js_data_tab. This function is used during the verification of part class lines, to retrieve values for distinct properties.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code
GET_WBK_TYPE

Description: This function is needed during BLL (bulkload log) analysis to fetch the workbook type from the log file contents. The function returns CDLST for codelists or OTHER for the rest, derived from the workbook name.

Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: CDLST for codelists; else OTHER

GET_XLS_ROW_NUMBER

Description: This function is needed during BLL (bulkload log) analysis to fetch the row number from the log file contents.

Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code

IDENT_ON_SHEET

Description: This function is an additional means of filtering the idents that belong to a sheet.

Type: Function
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code
**INIT_PARM_LIST**

Description: This procedure builds the table that is used to bind runtime variables to a retrieval source function call.

Type: Procedure

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code

**IS_GASKETCC**

Description: This function returns TRUE if the commodity code is a gasket.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: TRUE if gasket; else FALSE

**IS_NUTCC**

Description: This function returns TRUE if the commodity code is a nut.

Type: Function

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: TRUE if nut; else FALSE
### IS_WASHERCC

**Description:** This function returns TRUE if the commodity code is a washer.

**Type:** Function

**Usage:** SP3D 20.01

**Event:** Smart 3D Export Job

**Default:** TRUE if washer, else FALSE

### IS_PART_1SIZE_ONLY

**Description:** This function checks if the part has only one size component. If yes, it returns 1; else 0.

**Type:** Function

**Usage:** SP3D 20.01

**Event:** Smart 3D Export Job

**Default:** 1 if 1 size component; else 0

### MANIPULATE_MV_ITEM_IDENT_TEMP

**Description:** This procedure builds the table that is used to bind runtime variables to a retrieval source function call.

**Type:** Procedure

**Usage:** SP3D 20.01

**Event:** Smart 3D Export Job

**Default:** See CIP source code
MULTISIZE_ACTION_FOR_PCF

Description: This procedure is specific to the SDB configuration. The SecondSize attributes are set to NULL when the MultiSizeOption is present.
Assumption: MultiSizeOption - AC0030_001, SecondSizeFrom - AC0010_006, SecondSizeTo - AC0010_007, SecondSizeUnits - AC0010_008.

Type: Procedure
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code

OPEN_CPN_CUR

Description: This procedure opens a dynamic cursor if it is needed for building the catalog part number.

Type: Procedure
Usage: SP3D 40.21
Event: Import Job
Default: NULL

POST_PROCESS_FLTR_GEOM

Description: This procedure is called from SP3D 40.21 during the third stage of import after each part class sheet (sheet with category PART_DEFINITION) is loaded into a temporary table.

Type: Procedure
Usage: SP3D 40.21
Event: Import Job
Default: NULL
POST_PROCESS_SHT_DATA
Description: This procedure is called from SP3D 40.21 during the third stage of import after each part class sheet (sheet with category PART_DEFINITION) is loaded into a temporary table.
Type: Procedure
Usage: SP3D 40.21
Event: Import Job
Default: NULL

PRE_CHECK_FOR_SPEC_RULE_SHEETS
Description: This procedure checks if there is existing spec rule sheet data for a given spec based on the spec_code and SP_ID. If spec rule sheet data exists, this function simply returns. If no spec rule data exists, this procedure calls the fill_spec_rule_sheet method. To use this feature in the Smart 3D interface export, add the SP3D 10.13 procedure PRE_SPEC_RULE_TESTING to the Smart 3D spec rule sheet.
Type: Procedure
Usage: SP3D 20.01
Event: Smart 3D Export Job
Default: See CIP source code

RESET_SHT_CATEGORY
Description: This procedure sets the sheet category to PART DEFINITION.
Type: Procedure
Usage:
Event:
Default: UPDATE m_sp3d_shts SET Category = 'PART DEFINITION'
SAFE_TO_NUMBER

Description: This function safely converts strings to numbers. When the string is not able to be converted, the function returns NULL.

Type: FUNCTION

Usage:

Event: 

Default: to_number(txt)

SET_ATTR_FOR_S3D_VERSION

Description: This procedure is specific to the SDB configuration and is dependent on the SDB specific project default ZI_SP3D_VR. The procedure sets the XLS-IND in the sheet configuration as per the Smart 3D version selected in the project default. This function is called from the Smart 3D procedures on SP3D 10.13.

Type: Procedure

Usage: SP3D 10.13

Event: Click the Execute button.

Default: See CIP source code

SET_CATALOG_PART_NUMBER

Description: This procedure sets the company ident to the necessary values for Smart 3D.

Type: Procedure

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code
**SET_SHT_CATEGORY**

Description: This procedure sets the sheet category to GENERAL_LIST.

Type: Procedure

Usage:

Event:

Default: UPDATE m_sp3d_shts SET Category = 'GENERAL_LIST

**SHT_CUST_METHOD**

Description: This procedure fills the sheet data on the appropriate sheet in the export_job tables. It is called when the sheet retrieval method is SHT_CUST_METHOD.

Type: Procedure

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: NULL

**UPD_SI_SIZES_AND_SCHEDULES**

Description: This procedure updates sizes and schedules.

Type: Procedure

Usage: SP3D 20.01

Event: Smart 3D Export Job

Default: See CIP source code
Custom Packages

**M_PCK_SPEC_CUSTOM**

This package contains all customizable procedures and functions for specification management.

**CHECK_ISSUE_SPEC**

Description:

Type: Function

Usage: S.60.11

Event:

Default: NULL

**CHK_SPEC_RATING_GEOM**

Description: This function is called on S.50.06 if project default ZS_SRGH is set to CIP. Returns OK on success; otherwise, returns a message on failure.

Type: Function

Usage: S.50.06

Event:

Default: OK
**EXECUTE_CUSTOM_PROCEDURE**

**Description:** This procedure is called in SmartPlant Reference Data Plus if a customer defined procedure must be run. In addition to the PL/SQL custom logic within this procedure, if a .NET custom method needs to be called, then the OUT parameters `p_assembly_name`, `p_assembly_type`, `p_class_name`, `p_method_name`, `p_parameter_tab`, `p_parameter_datatype` need to be specified.

- `p_from_region` - Specifies the region from which a call to this procedure was made ('SpecGeneralArea' if called from Spec General Area)
- `p_assembly_name` - Specify the name of the .NET assembly. The assembly should reside in the same directory where the SmartPlant Reference Data.exe resides.
- `p_assembly_type` - Specify the type of the .NET assembly ('dll' or 'exe').
- `p_class_name` - Specify the name of the class within the assembly.
- `p_method_name` - Specify the name of the method within the class to be invoked.
- `p_parameter_tab` - Specify the parameters values for the method. The index should start from 1.
- `p_parameter_datatype_tab` - Specify the data type of the parameter of the .NET custom method. Valid values are 'Single', 'Double', 'Decimal', 'String', 'Long', 'Integer', 'Boolean', and 'Date'. The index should start from 1.

**Type:** Procedure

**Usage:** SmartPlant Reference Data Plus

**Event:**

**Default:** See CIP source code

---

**GENERAL_CIP**

**Description:** This function is called when the Execute CIP button is clicked on the S.50.06 Maintain Specifications screen.

**Type:** Function

**Usage:** S.50.06

**Event:** Click the Execute CIP button.

**Default:** None
**GENERATE_SPEC_ITEMS**

Description: This procedure is called in SmartPlant Reference Data Plus if the generation method for spec items is set to CIP or Standard and CIP.

Type: Procedure

Usage: SmartPlant Reference Data Plus

Event: NULL

**RELEASE_SPEC**

Description:

Type: Function

Usage:

Event: NULL

**M_PCK_SPEC_REV_CUSTOM**

This package contains all customizable procedures and functions for specification revision management.

**BUILD_RELEASE_LIST**

Description:

Type: Procedure

Usage: S.50.13

Event:

Default: See CIP source code
CHECK_FOR_UNRELEASED

Description:
Type: Function
Usage: S.50.06, S.50.13
Event:
Default: See CIP source code

CHECK_TEXT

Description:
Type: Function
Usage: S.50.06, S.50.13
Event:
Default: 0

DELETE_SPEC_IDENTS

Description: This procedure is executed from S.50.13 when a spec is unissued. It deletes records from m_released_spec_idents.
Type: Procedure
Usage: S.50.13
Event: Unissue Spec
Default: DELETE FROM m_released_spec_idents
Custom Packages

**INSERT_SPEC_IDENTS**

Description: This procedure is executed from S.50.06 or S.50.13 when a spec is issued. It inserts records into `m_released_spec_idents`.

Type: Procedure

Usage: S.50.06, S.50.13

Event: Issue Spec

Default: See CIP source code

**NEW_REV_GEOM_NAME**

Description: This function creates new geometric names when a spec is revised.

Type: Function

Usage: S.50.06, S.50.13

Event: Revise spec

Default: See CIP source code

**M_PCK_SPISO_CUSTOM**

This custom package contains functions and procedures to support the export of specification data to SmartPlant Isometrics.

**ComponentTypeForBendPipe**

Description: This function returns the component type for the bends.

Type: Function

Usage: SPRD Plus XI3020 Export Jobs

Event: Export specification

Default: 5D
CONVERT_BORE_UNIT_TO_UNIT

Description: This function is executed when the export unit is set for an export job on the SPIso Export Jobs screen. The purpose is to convert all bore values from source unit to target unit. The used unit codes for conversion may vary depending on the customer's database. Here unit codes IN and MM are used to identify the correct S.40.04.xx conversion table.

Parameters passed in:
- p_from_unit_id - current unit ID to be converted
- p_to_unit_id - identifies the unit ID for output
- p_value - value to be converted

This code is not maintained by Intergraph; as with all CIP code, it is the customer's responsibility to use this API and maintain their custom code. Customers must save their code as text files, because SmartPlant Reference Data and SmartPlant Materials database migration overwrites it without any warning.

**IMPORTANT** This function must not commit. Otherwise, the export job may not work properly.

Type: Function
Usage: SPRD Plus XI3020 Export Jobs
Event: Export specification
Default: The input value converted to the target unit
CONVERT_DIM_UNIT_TO_UNIT

Description: This function is executed when the export unit is set for an export job on the SPIso Export Jobs screen. The purpose is to convert all dimensional values from source unit to target unit. The used unit codes for conversion may vary depending on the customer's database. Here unit codes IN and MM are used to identify the correct S.40.04.xx conversion table.

Parameters passed in:
- p_from_unit_id - current unit ID to be converted
- p_to_unit_id - identifies the unit ID for output
- p_value - value to be converted

This code is not maintained by Intergraph; as with all CIP code, it is the customer's responsibility to use this API and maintain their custom code. Customers must save their code as text files, because SmartPlant Reference Data and SmartPlant Materials database migration overwrites it without any warning.

**IMPORTANT** This function must not commit. Otherwise, the export job may not work properly.

Type: Function
Usage: SPRD Plus XI3020 Export Jobs
Event: Export specification
Default: The input value converted to the target unit

EXPORT_FILE_NAME

Description: This function is used to set the name of the export file.

Type: Function
Usage: SPRD Plus XI3020 Export Jobs
Event: Export specification
Default: Spec code
FORMAT_BOLT_CC

Description: This function is used to format a bolt commodity code.

Type: Function

Usage: SPRD Plus XI3020 Export Jobs

Event: Export specification

Default: Bolt commodity code

GENERAL_BEND

Description: This procedure is called after the bend data for an export job has been collected, but before the export files are generated.

Parameters passed in:

- p_spiej_id - Primary key of table m_spiso_export_jobs; this key identifies the current job uniquely.
- p_spies_id - Primary key of table m_spiso_export_specs.

This code is not maintained by Intergraph; as with all CIP code, it is the customer’s responsibility to use this API and maintain their custom code. Customers must save their code as text files, because SmartPlant Reference Data and SmartPlant Materials database migration overwrites it without any warning.

**IMPORTANT** This function must not commit. Otherwise, the export job may not work properly.

Type: Procedure

Usage: SPRD Plus XI3020 Export Jobs

Event: Export specification

Default: NULL
Custom Packages

**GENERAL_BRANCH**

**Description:** This procedure is called after the branch data for an export job has been collected, but before the export files are generated.

Parameters passed in:

- `p_spiej_id` - Primary key of table `m_spiso_export_jobs`; this key identifies the current job uniquely.
- `p_spies_id` - Primary key of table `m_spiso_export_specs`.

This code is not maintained by Intergraph; as with all CIP code, it is the customer's responsibility to use this API and maintain their custom code. Customers must save their code as text files, because SmartPlant Reference Data and SmartPlant Materials database migration overwrites it without any warning.

**IMPORTANT:** This function must not commit. Otherwise, the export job may not work properly.

**Type:** Procedure

**Usage:** SPRD Plus XI3020 Export Jobs

**Event:** Export specification

**Default:** NULL

**GENERAL_CIP**

**Description:** This procedure is executed after collecting data for an export job in the export tables, but prior to generation of export data.

Parameter passed in:

- `p_spiei_id` - Primary key of table `m_spiso_export_jobs`; this key identifies the current job uniquely.

For a spec driven export, `spiei_id` permits the selection of specs attached to the job (`pk_id = spies_id`).

This code is not maintained by Intergraph; as with all CIP code, it is the customer's responsibility to use this API and maintain their custom code. Customers must save their code as text files, because SmartPlant Reference Data and SmartPlant Materials database migration overwrites it without any warning.

**IMPORTANT:** This function must not commit. Otherwise, the export job may not work properly.

**Type:** Procedure
**GENERAL_DEFAULT_CHOICE**

**Description:** An export job fills the `M_SPISO_EXP_DEFAULT_CHOICES` table. Afterwards, this procedure is executed to update the stored records in this table, according to your requirements. And then, the modified data is exported to the `DEFAULTCHOICE` table in the SmartPlant Isometrics MDB file.

**Parameters passed in:**
- `p_spiej_id` - Primary key of table `m_spiso_export_jobs` that uniquely identifies the current job.
- `p_spec_header_id` - Primary key of table `m_spec_headers`.

This code is not maintained by Intergraph; as with all CIP code, it is the customer's responsibility to use this API and maintain their custom code. Customers must save their code as text files, because SmartPlant Reference Data and SmartPlant Materials database migration overwrites it without any warning.

**IMPORTANT:** This function must not commit. Otherwise, the export job may not work properly.

**Type:** Procedure

**Usage:** SPRD Plus XI3020 Export Jobs

**Event:** Export specification

**Default:** Update of `M_SPISO_EXP_DEFAULT_CHOICES` table (see source code)
**GENERAL_GET_BOLT**

**Description:** This procedure is executed after collecting data for an export job in the export tables, but prior to generation of export data.

**Parameters passed in:**
- **p_spiej_id** - Primary key of table `m_spiso_export_jobs`; this key identifies the current job uniquely.
  - For a spec driven export, `spiej_id` permits the selection of specs attached to the job (pk_id = spies_id).
- **p_bolt_code_length**

  This code is not maintained by Intergraph; as with all CIP code, it is the customer’s responsibility to use this API and maintain their custom code. Customers must save their code as text files, because SmartPlant Reference Data and SmartPlant Materials database migration overwrites it without any warning.

  **IMPORTANT:** This function must not commit. Otherwise, the export job may not work properly.

**Type:** Procedure

**Usage:** SPRD Plus XI3020 Export Jobs

**Event:** Export specification

**Default:** NULL (Sample code available)

---

**GET_BORE_VALUE_CONV**

**Description:** This function is used to find the attribute value for the ident and unit.

**Type:** Function

**Usage:** SPRD Plus XI3020 Export Jobs

**Event:** Export specification

**Default:** See source code
GET_CC_DESCRIPTION
Description: The purpose of this function is to fill a user-definable export value for the commodity code description. The returned value is truncated to 512 characters. This function is executed by m_pck_spiso_export during export for each commodity code.
Parameter passed in:
   p_cc_id - commodity_id from table m_commodity_codes
Type: Function
Usage: SPRD Plus XI3020 Export Jobs
Event: Export specification
Default: Substr(m_commodity_code_nls.short_desc,1,512)

GET_DIM_VALUE_CONV
Description: This function is used to find the attribute value for the ident and unit.
Type: Function
Usage: SPRD Plus XI3020 Export Jobs
Event: Export specification
Default: See source code

GET_IDENT_BOLT_DIMS
Description: This function is used to derive attribute values for bolt diameter, bolt length, and bolt quantity (BDIA, BLGT, BQTY).
Parameters passed in:
   - p_ident - Current ident to export
   - p_interface_id - Stored for the SPIso interface in table m_interfaces
   - p_translation - Fixed attribute translations (BDIA, BLGT, BQTY) passed in
Type: Function
Usage: SPRD Plus XI3020 Export Jobs
GET_IDENT_DESCRIPTION

Description: The purpose of this function is to fill a user-definable export value for the ident description. The returned value is truncated to 512 characters. This function is executed by m_pck_spiso_export during export for each ident.

Parameter passed in:
- p_ident - Ident from table m_idents

Type: Function

Usage: SPRD Plus XI3020 Export Jobs

Event: Export specification

Default: Substr(m_commodity_code_nls.short_desc||m_ident_nls.short_desc,1,512)

GetIdentWeight

Description: This function can be used to get the weight of the ident.

Type: Function

Usage: SPRD Plus XI3020 Export Jobs

Event: Export specification

Default: 0
**SPEC_IDENTIFIER**
Description: This function is used to set the internal identifier of the export file.
Type: Function
Usage: SPRD Plus XI3020 Export Jobs
Event: Export specification
Default: Substr(spec code, 1, 50)

**M_PCK_STD_CUSTOM**
This custom package contains functions and procedures used in SmartPlant Reference Data.

**CC_DESC**
Description: This function returns the commodity code description.
Type: Function
Usage: S.50.06
Event:
Default: mvp_commodity_code_layouts.layout_short

**CHECK_BRANCH**
Description: This function returns the branch table for the entered short code.
Type: Function
Usage: S.50.06
Event:
Default: m_geom_branches.gb_code
CHECK_INV_IDENTS
Description: This function checks if specification item idents are valid.
Type: Function
Usage: PDMS.20.01
Event: See CIP source code

CHECK_SHG
Description: This function returns the spec header geometric table for the entered short code.
Type: Function
Usage: S.50.06
Event: mvp_geom_names.gn_code

CMS_IDENT_CODE
Description: This function returns the ident code for the Component Management System.
Type: Function
Usage: XC.20.G.47
Event: CMS_I||m_idents.ident
CMS_IDENT_PROP_VAL_POST_INSUPD

Description: This procedure is called after an ident property value is created or updated.
Type: Procedure
Usage: XC.20.G.47
Event: 
Default: NULL

CODE_BLD_CIP

Description: This procedure is called at the end of the code_bld procedure in M_PCK_MLCL.
Type: Procedure
Usage: S.50.06
Event: 
Default: NULL

COMMODITY_CODE_CHECK

Description: This function is called before insert or update of a commodity code and can be used to perform customer-defined checks.
Values of parameter p_mode: I=Input, U=Update
p_error can be used to return an error message to the user.
Type: Function
Usage: S.30.01, S.30.11
Event: Commodity code insert or update
Default: TRUE
CREATE_SUPPLIER

Description: This procedure is executed when a new supplier is created for a company.
Type: Procedure
Usage: A.10.23, A.10.24, OSCI.10.01
Event: Create supplier
Default: See CIP source code

GET_C_IDENT

Description: This function retrieves the company ident code. If no company ident code exists, the function returns the ident code. This function is only used when you are working with company idents.
Type: Function
Usage: PDS.10.10, PDS.10.12, S.50.06
Event:
Default: MVP_IDENT_COMPANIES.ic_code

GET_IDENT

Description: This function returns the ident for a given company ident code.
Type: Function
Usage: AS.10.03, OPI.10.01, S.50.06
Event:
Default: mvp_idents.ident
GET_IDENT_NLS

Description: This function returns the ident description.

Type: Function

Usage:

Event:

Default: See CIP source code

GET_IDENT_NLS_TRANS

Description: This function returns the translated value of the ident description.

Type: Function


Event:

Default: See CIP source code
GET_PROJ_ID

Description:  This function returns the project (proj_id) of the current session.

Type: Function

Usage:  S.50.06, S.80.01

Event:  Default:  mpck_login.proj_id

IDENT_DESC

Description:  This function returns the description of an ident.

Type: Function


Event:  Default:  ident_desc(p_ident, mpck_login.nls_id)
**IDENT_DESC_HIST**

Description: This function returns the ident short description from the ident history table.

Type: Function

Usage: R.30.01

Event:

Default: `m_ident_nls_hists.short_desc`

**IDENT_LONG_DESC**

Description: This function returns the long description of an ident.

Type: Function

Usage: C.20.01.02

Event:

Default: See CIP source code

**NEW_IDENT_CODE**

Description: This function creates the ident code for a new ident.

Type: Function

Usage: B.20.01, B.20.02, B.20.03, C.20.01, C.20.01.02, C.20.03, C.20.04, P.30.22, P.30.25, P.30.27, P.50.07, P.70.31, P.70.62, P.70.72, P.80.11, P.80.51, R.30.01, S.30.01, S.30.11, S.80.01

Event: Ident creation

Default: `ident_code='I'||m_idsents.ident`
NEW_RB_IDENT_CODE
Description: This function creates the ident code for a new ident.
Type: Function
Usage: S.50.06, S.80.01
Event: Ident creation
Default: 'I' || p_ident_number

NUMBER_DN
Description: This function returns the number of nominal sizes of an ident.
Type: Function
Usage: S.50.06
Event:
Default: See CIP source code

NUMBER_SCHEDULE
Description: This function returns the number of schedules of an ident.
Type: Function
Usage: S.50.06
Event:
Default: See CIP source code
POST_BUILD_IDENTS_CIP

Description: This procedure is called by m_pck_build_idents.mp_build_spec_idents during ident creation at a point in time when obsolete idents are already removed but the ccm_text flag on table m_idents is still set to 'NEW'. Together with the input parameter commodity_id, the affected idents can be identified.

Note: To avoid any later conflicts with displayed ident codes, be sure to keep ident codes unique with respect to index m_idents.m_idents_N02 (proj_id, ident_code, rev_end).

Type: Procedure
Usage: S.50.06
Event: Build idents
Default: NULL

POST_FINAL_BUILD_IDENTS_CIP

Description: This procedure is called by m_pck_build_idents.build_idents at the end of the procedure.

Type: Procedure
Usage: S.50.06
Event: Build idents
Default: NULL (Sample available)

S5007_INFORMATIONS

Description:

Type: Procedure
Usage: S.50.06
Event:
Default: See CIP source code
Custom Packages

M_PCK_TAG_RUL_CUSTOM
This package implements all functionality for processing a tag rule definition.

UPDATE_CC_WITH_TAG_DESC
Description: This procedure updates the commodity code layout. The procedure is executed during the BOM import.
Type: Procedure
Usage: B.40.01
Event: BOM import
Default: See CIP source code

UPD_TAGDESC_WITH_RLI_COMMENT
Description: This procedure updates the commodity code layout. The procedure is executed when you click the Update Tag Desc button on Window 2 of R.30.01 Maintain Requisitions.
Type: Procedure
Usage: R.30.01
Event: Click the Update Tag Desc button.
Default: See CIP source code

M_PCK_TRAFFIC_CUSTOM
This package contains all customizable procedures and functions for traffic.
**CALC_SELL_PRICES**

Description: Allocates the selling price of the release note to the package items. This procedure is called from the **P.70.74 Packages** screen when the **Populate** button is clicked on the **With Release Notes** tab.

Type: Procedure

Usage: P.70.74

Event: Click the **Populate** button.

Default: NULL

**ERP_TRANSFER**

Description: Performs ERP transfer. This procedure is called from the **P.70.74 Packages** screen when the **ERP Transfer** button is clicked on the **With Release Notes** tab.

Type: Procedure

Usage: P.70.74

Event: Click the **ERP Transfer** button.

Default: UPDATE m_release_notes SET erp_set_ind = 'Y'

**GET_SUPP_LABEL**

Description: This function returns the supplement label to be shown within the traffic tree on the **PT.20.01** screen.

Type: Function

Usage: PT.20.01

Event: Open screen

Default: 'Suppl ' || TO_CHAR(tfr_supp - 1) || ' - ' ||TO_CHAR(freeze_date, 'DD-MON-YYYY') from m_traffic_freezes
**MOC_LABEL**

Description: This function returns the label to be shown for a means of conveyance within the traffic tree on the PT.20.01 screen.

Type: Function

Usage: PT.20.01

Event: Open screen

Default: m_cargo_ships.carrier_name

**REVERT_ERP_TRANSFER**

Description: Reverts ERP transfer. This procedure is called from the P.70.74 Packages screen when the Revert ERP Transfer button is clicked on the With Release Notes tab.

Type: Procedure

Usage: P.70.74

Event: Click the Revert ERP Transfer button.

Default: UPDATE m_release_notes SET erp_set_ind = 'N'

**M_PCK_UOM_CUSTOM**

This package contains all customizable procedures and functions for the UOM feature.

**GET_VALUE_ASS**

Description: The process of building assembly idents on S.70.01 Build Assembly Structure uses this function to convert the size range values of the assembly positions.

Type: Function

Usage: S.70.01

Event: Building assembly idents
M_PCK_VDR_CUSTOM

This package contains all customizable procedures and functions for the VDR screens.

GEN_DOC_NUMBER

Description: This function is called by P.70.81 VDR Submittal and Status when a new record on D.10.11 Documents is created. The function generates a document code.

Type: Function
Usage: P.70.81
Event: Entering and saving a client reference name for a document
Default: ''

GEN_REFERENCE

Description: This function is called by P.70.81 VDR Submittal and Status when a new record on D.10.11 Documents is created. This function generates a value for 'Reference'.

Type: Function
Usage: P.70.81
Event: Automatic creation of a document on D.10.11 (occurs when just entering a client reference name) or when entering and saving a client reference name
Default: ''
Custom Packages

**GET_ADDR_TYPE**

Description: This function returns the address type that is to be used on the Agreement Details window of the P.70.81 VDR Submittal and Status screen.

Type: Function

Usage: P.70.81

Event: Open the Agreement Details window of P.70.81

Default: NULL (sample code available)

**GET_CMT_EMAIL**

Description: This function returns the communication type that represents 'Email'.

Type: Function

Usage: P.70.81

Event: Open the Agreement Details window of P.70.81.

Default: NULL (sample code available)

**GET_CMT_FAX**

Description: This function returns the communication type that represents 'Fax'.

Type: Function

Usage: P.70.81

Event: Open the Agreement Details window of P.70.81.

Default: NULL (sample code available)
GET_CUSTOM_FIELD

Description: This function is called to fill the custom fields on the P.70.81 VDR Submittal and Status screen.

Type: Function
Usage: P.70.81
Event: Open screen
Default: NULL (sample code available)

GET_CUSTOM_PROMPT

Description: This function is called to get the labels for the custom fields on the P.70.81 VDR Submittal and Status screen.

Type: Function
Usage: P.70.81
Event: Open screen
Default: NULL (sample code available)

GET_EXPED_ADDR_TYPE

Description: This function returns the address type for the expediting contact that is to be used on the Agreement Details window of the P.70.81 VDR Submittal and Status screen.

Type: Function
Usage: P.70.81
Event: Open the Agreement Details window of P.70.81.
Default: NULL (sample code available)
Custom Packages

GET_FORMAT_ATTR_ID
Description: This function returns the primary key of the attribute the valid values of which are to be used for the LOV in the Format field on the P.70.81 VDR Submittal and Status screen.
Type: Function
Usage: P.70.81
Event: Open the Agreement Details window of P.70.81.
Default: NULL

GET_PO_ATTR_ID
Description: This function returns the primary key of the attribute that is to be displayed on the P.70.81 VDR Submittal and Status screen. This attribute is or will be assigned to the agreement (line item).
Type: Function
Usage: P.70.81
Event: Open the Agreement Details window of P.70.81.
Default: NULL

GET_REVIEWER
Description: This function returns the reviewer for a distribution category on the A.10.52 Distribution Categories screen.
Type: Function
Usage: A.10.52
Event: Querying records in second block
Default: NULL
**REMOVE_CONTENT**

**Description:** This procedure is called when the Approval Code field is filled on the P.70.81 VDR Submittal and Status screen.

**Type:** Procedure

**Usage:** P.70.81

**Event:** Setting the approval code

**Default:** NULL

**RETURNS_REPORT**

**Description:** This function returns the name of the report that is started when the Returns button is clicked on the P.70.81 VDR Submittal and Status screen.

**Type:** Function

**Usage:** P.70.81

**Event:** Click the Returns button.

**Default:** 'P70R88'

**RETURN_TRANSMITTAL**

**Description:** This procedure is executed when the Returns button is clicked on the P.70.81 VDR Submittal and Status screen.

**Type:** Procedure

**Usage:** P.70.81

**Event:** Click the Returns button.

**Default:** NULL
Custom Packages

REV_REFERENCEx
Description: This function is called by P.70.81 VDR Submittal and Status when a document is revised. The return value is used to update the value in the Reference field.

Type: Function
Usage: P.70.81
Event: Revise document
Default: reference_module

SET_FIELD_VALUES
Description: This procedure is called after all screen inserts or updates have been performed on the P.70.81 VDR Submittal and Status screen before data is finally committed.

Type: Procedure
Usage: P.70.81
Event: Commit data
Default: NULL

VDR_TRANSMITTAL
Description: This procedure is called when the VDR Transmittal button is clicked on the P.70.81 VDR Submittal and Status screen.

Type: Procedure
Usage: P.70.81
Event: Click the VDR Transmittal button.
Default: NULL
M_PCK_VERIFY_CUSTOM

This package provides a way to hook customer-defined PL/SQL procedures into a B.20.01.41 verification job.

BLD_AMBLY

Description: This procedure creates assembly positions.
Type: Procedure
Usage: B.20.01.41
Event: Start button is clicked, when this list check is attached
Default: See CIP source code

BLD_ASMBLY

Description: This procedure creates assembly positions.
Type: Procedure
Usage: B.20.01.41
Event: Start button is clicked, when this list check is attached
Default: See CIP source code

CHK_BRANCH

Description: Re-creates the sub position of a list position with current values from spec branch tables.
Type: Procedure
Usage: B.20.01.41
Event: Called by find_ident check if project default ZB_FI_BRNC = ‘Y’
Default: m_pck_bom_utl.chk_branch(p_lp_id)
EE_CHECK
Description: Calls a function that checks the validity of the spec, commodity code, ident, group, and part information stored on a list position.
Type: Procedure
Usage: B.20.02
Event: Dummy to be called from other CIPs of this package.
Default: m_pck_easy_entry.check_pos(p_lp_id)

FIND_SPEC_ASS
Description:
Type: Function
Usage: B.20.01.41
Event: Called by list check BLD_ASMBLY.
Default: See CIP source code

FINISH_JOB
Description: This procedure finishes a job.
Type: Procedure
Usage: B.20.01.41
Event: Called after a job stops processing.
Default: See CIP source code
INIT_JOB
Description: This procedure initializes a job.
Type: Procedure
Usage: B.20.01.41
Event: Called before a job starts processing.
Default: See CIP source code

NULLIFY_LIST_POS
Description: This function is called by the find_ident standard verification job when find_ident does not find an ident. Should the old ident, commodity, and so on be removed?
Type: Function
Usage: B.20.01.41
Event: Find ident
Default: FALSE if short_code is NULL; otherwise, TRUE

TEST_ASSMB_UNIQUE_UNIT_SYSTEM
Description: 
Type: Function
Usage: B.20.01.41
Event: Called by list check BLD_ASMBLY.
Default: See CIP source code