Robot Open Standard

version 8.1

Table of Contents

ROBOT OBJECT MODEL 20
GLOBAL DATA TYPES 21
IRobotValuesArray *21*
IRobotNamesArray *21*
IRobotNumbersArray *22*
IRobotObjectsArray *22*
IRobotNumbersCollection *23*
IRobotCodeRegistrar *23*
IRobotWindow *24*
IRobotWindowState *24*
IRobotProtectionInfo *25*
IRobotComponentFactory *25*
IRobotComponentType *25*
IRobotDegreeOfFreedom *30*
IRobotPointsArray *30*
IRobotSerializable *31*
GEOMETRIC DATA TYPES 31
IRobotGeoPoint2D *31*
IRobotGeoPoint3D *32*
IRobotGeoCoordinateAxis *32*
IRobotGeoCoordinateSystem *32*
IRobotGeoSegmentType *33*
IRobotGeoObjectType *33*
IRobotGeoCurveDiv *34*
IRobotGeoSegment *34*
IRobotGeoSegmentLine *34*
IRobotGeoSegmentArc *35*
IRobotGeoObject *35*
IRobotGeoPolyline *35*
IRobotGeoSegmentCollection *36*
IRobotGeoContour *36*
IRobotGeoArc *36*
IRobotGeoCircle *36*
IRobotGeoPoint3DCollection *37*
IRobotGeoLayer *37*
IRobotGeoArcDefinitionMethod *38*
IRobotGeoCoordinateAxisSense *38*
IRobotNumbersDictionary *39*
APPLICATION - MAIN OBJECT OF THE MODEL 39
IRobotApplication *39*
IRobotPreferences *41*
IRobotDirectory *41*
IRobotLanguage *42*
IRobotQuitOption *42*
IRobotDirectoryExtension *42*
IRobotPreferencesEvents *43*
PROJECT 43
IRobotProjectType *43*
IRobotProject *44*
IRobotProjectEvents *47*
IRobotFileInsertParams *47*
IRobotProjectSaveFormat *48*
JOB PREFERENCES 49
IRobotProjectPreferences *49*
IRobotSectionDatabaseList *50*

IRobotMaterialDatabase *51*
IRobotCodeType *52*
IRobotSectionDatabase *52*
IRobotProjectPreferencesEvents *53*
UNITS AND FORMATS 53
IRobotUnitType *53*
IRobotUnitData *55*
IRobotUnitEditionType *55*
IRobotUnitMngr *56*
IRobotUnitComplexData *57*
IRobotUnitEditionServer *57*
IRobotUnitEditionData *58*
IRobotUnitMngrEvents *58*
PROJECT COMPONENTS 58
IRobotProjectComponentType *58*
IRobotProjectComponent *59*
IRobotProjectComponentMngr *60*
CONCRETE FOOTING 60
IRConcrFooting *60*
IRConcrFootingGeometry *61*
IRConcrFootingDimType *62*
IRConcrFootingShapeType *63*
IRConcrFootingType *63*
IRConcrFootingGround *63*
IRConcrFootingLoads *64*
IRConcrFootingResults *64*
SUPPORT OF EXTERNAL FILE FORMATS 64
IRobotExternalFileFormat *64*
IRobotSTRParams *65*
IRobotSTRParameter *65*
IRobotSTRParamType *66*
DATA SERVER 67
IRobotStructure *67*
IRobotStructureApplyInfo *69*
IRobotStructureCache *69*
IRobotStructureMergeData *70*
BASIC DEFINITIONS 70
UNIFORM ACCESS TO STRUCTURE COMPONENTS 71
IRobotDataObject *71*
IRobotDataObjectServer *72*
IRobotObjectType *73*
QUERYING MECHANISM 74
IRobotSelection *74*
IRobotCollection *75*
IRobotMultiSelection *76*
IRobotMultiCollection *76*
IRobotModeSelection *77*
LABELS - COMPLEX ATTRIBUTES OF OBJECTS 77
IRobotLabelType *78*
IRobotLabel *80*
IRobotLabelServer *80*
SELECTIONS 82
IRobotSelectionFactory *82*
IRobotModeSelectionType *83*
IRobotModeCombinationType *83*
IRobotPredefinedSelection *84*

IRobotGroup *84*
IRobotGroupServer *85*
STRUCTURE COMPONENTS 86
NODES 86
IRobotNode *86*
IRobotNodeServer *87*
COMPLEX ATTRIBUTES OF A NODE 89
IRobotNodeSupportData *89*
IRobotNodeSupportFixingDirection *91*
IRobotNodeSupportOneDirectionFixingType *91*
IRobotEmitter *91*
IRobotAdvancedSupportType *91*
RIGID LINKS 92
IRobotNodeRigidLinkData *92*
IRobotNodeRigidLinkDef *93*
IRobotNodeRigidLinkServer *93*
COMPATIBLE NODES 94
IRobotNodeCompatibilityData *95*
IRobotNodeCompatibilityDef *96*
IRobotNodeCompatibilityServer *97*
BARS 98
IRobotBar *98*
IRobotBarServer *101*
IRobotBarEnd *103*
IRobotBarTensionCompression *103*
IRobotBarElement *104*
COMPLEX ATTRIBUTES OF A BAR 104
IRobotBarOffsetData *104*
IRobotBarEndOffsetData *105*
IRobotBarEndBracketType *105*
IRobotBarEndBracketDataValue *105*
IRobotBarEndBracketData *105*
IRobotBarOffsetAutoPosition *106*
IRobotBarOffsetMemberLength *107*
SECTIONS 107
IRobotBarSectionShapeType *107*
IRobotBarSectionDataValue *113*
IRobotBarSectionType *115*
IRobotBarSectionNonstdDataValue *116*
IRobotBarSectionConcreteDataValue *118*
IRobotBarSectionConcreteCutsPosition *120*
IRobotBarSectionData *120*
IRobotBarSectionNonstdData *123*
IRobotBarSectionConcreteData *123*
IRobotBarSectionElasticParams *124*
IRobotBarSectionComplexData *125*
IRobotBarSectionComponentShape *126*
CABLES 127
IRobotBarCableData *127*
IRobotBarCableAssemblingParamType *127*
ELASTIC GROUND 128
IRobotBarElasticGroundData *128*
RELEASES 128
IRobotBarEndReleaseValue *128*
IRobotBarReleaseData *129*
IRobotBarEndReleaseData *129*

GEOMETRICAL IMPERFECTIONS 131
IRobotBarGeoImperfectionsData *131*
IRobotBarGeoImperfectionsAxis *131*
NON-LINEAR HINGES 132
IRobotNonlinearHingeModelServer *132*
IRobotNonlinearHingeModel *132*
IRobotNonlinearHingeModelAxisParams *134*
IRobotNonlinearHingeModelType *134*
IRobotNonlinearHingeModelPoints *135*
IRobotNonlinearHingeModelUnloadingType *135*
IRobotNonlinearHingeData *136*
IRobotNonlinearHingeComponentType *136*
IRobotNonlinearHingeDef *137*
IRobotNonlinearHingeServer *137*
CLADDING 138
IRobotCladdingData *138*
IRobotCladdingType *139*
IRobotCladdingMethod *139*
LOAD CASES 139
IRobotCase *140*
IRobotCaseNature *140*
IRobotCaseAnalizeType *141*
IRobotCaseType *142*
IRobotCaseServer *143*
IRobotSimpleCase *144*
IRobotCaseFactorMngr *145*
IRobotCaseCombination *146*
IRobotCombinationType *147*
IRobotCaseFactor *148*
IRobotCaseCollection *148*
IRobotLimitState *148*
IRobotCaseAnalysisModesFilter *149*
IRobotCaseAnalysisModesFilterType *149*
IRobotDynamicAnalysisExcitationDirection *150*
LOAD RECORDS 150
IRobotLoadRecordType *150*
IRobotLoadRecord *152*
IRobotLoadRecordMngr *152*
IRobotLoadRecordLinear3D *153*
IRobotLoadRecordIn3Points *153*
IRobotLoadRecordThermalIn3Points *154*
IRobotLoadRecordInContour *155*
IRobotLoadRecordLinear *156*
IRobotLoadRecord2 ..... *156*
VALUES DESCRIBING LOAD RECORDS 156
IRobotNodeForceRecordValues *157*
IRobotNodeDisplacementRecordValues *157*
IRobotNodeAuxiliaryRecordValues *157*
IRobotBarDilatationRecordValues *158*
IRobotBarForceConcentrateRecordValues *158*
IRobotBarMomentDistributedRecordValues *158*
IRobotBarUniformRecordValues *159*
IRobotBarTrapezoidaleRecordValues *159*
IRobotBarThermalRecordValues *160*
IRobotBarDeadRecordValues *160*
IRobotPointAuxiliaryRecordValues *160*

IRobotIn3PointsRecordValues *161*
IRobotPressureRecordValues *161*
IRobotLinearRecordValues *162*
IRobotMassActivationRecordValues *162*
IRobotLinear3DRecordValues *162*
IRobotThermalIn3PointsRecordValues *163*
IRobotInContourRecordValues *163*
IRobotUniformRecordValues *164*
IRobotThermalRecordValues *164*
IRobotNodeForceInPointRecordValues *165*
IRobotLinearOnEdgesRecordValues *166*
IRobotDeadRecordValues *166*
IRobotSurfaceOnObjectRecordValues *167*
IRobotMobilePointForceRecordValues *167*
IRobotMobileDistributedRecordValues *168*
MODAL ANALYSIS PARAMETERS 169
IRobotModalAnalysisMode *169*
IRobotModalAnalysisAlgorithm *169*
IRobotModalAnalysisMassMatrixType *170*
IRobotModalAnalysisLimitType *170*
IRobotModalAnalysisParams *171*
IRobotModalAnalysisBase *172*
IRobotModalAnalysisLimits *173*
IRobotModalAnalysisShifts *173*
IRobotMassEccentricities *174*
MODAL ANALYSIS PARAMETERS TAKING INTO ACCOUNT STATIC FORCES 174
IRobotModalWithStaticForcesAnalysisParams *174*
SEISMIC ANALYSIS PARAMETERS 175
IRobotSeismicAnalysis_AFPS_90_Params *175*
IRobotSeismicAnalysis_CHINESE_Params *176*
IRobotSeismicAnalysis_CIRSOC_103_Params *177*
IRobotSeismicAnalysis_DM_16_1_96_Params *177*
IRobotSeismicAnalysis_EC_8_Params *178*
IRobotSeismicAnalysis_P_100_92_Params *179*
IRobotSeismicAnalysis_PS_69_Params *179*
IRobotSeismicAnalysis_PS_92_Params *180*
IRobotSeismicAnalysis_RPA_88_Params *181*
IRobotSeismicAnalysis_TURKISH_23098_Params *182*
IRobotSeismicAnalysis_UBC_97_Params *182*
IRobotSeismicAnalysis_IBC_2000_Params *183*
IRobotSeismicAnalysis_EAK_2000_Params *184*
IRobotSeismicAnalysis_AFPS_90_ZoneType *185*
IRobotSeismicAnalysis_CIRSOC_103_ZoneType *185*
IRobotSeismicAnalysis_EC_8_ZoneType *186*
IRobotSeismicAnalysis_P_100_92_ZoneType *186*
IRobotSeismicAnalysis_PS_92_ZoneType *187*
IRobotSeismicAnalysis_RPA_88_ZoneType *187*
IRobotSeismicAnalysis_UBC_97_ZoneType *188*
IRobotSeismicAnalysis_EAK_2000_ZoneType *188*
IRobotSeismicAnalysis_TURKISH_23098_ZoneType *189*
IRobotSeismicAnalysis_AFPS_90_StructureType *189*
IRobotSeismicAnalysis_AFPS_90_SiteType *190*
IRobotSeismicAnalysisSpectrumType *190*
IRobotSeismicAnalysis_CHINESE_StructureType *190*
IRobotSeismicAnalysis_CHINESE_SiteType *191*
IRobotSeismicAnalysis_CHINESE_IntensityType *191*

IRobotSeismicAnalysis_CHINESE_DesignType *192*
IRobotSeismicAnalysis_CHINESE_EarthquakeType *192*
IRobotSeismicAnalysis_CIRSOC_103_SoilType *192*
IRobotSeismicAnalysis_CIRSOC_103_StructureType *193*
IRobotSeismicAnalysisDirectionType *193*
IRobotSeismicAnalysis_DM_16_1_96_ProtectionCoeffType *193*
IRobotSeismicAnalysis_EC_8_StructureType *194*
IRobotSeismicAnalysis_EC_8_SoilType *194*
IRobotSeismicAnalysis_P_100_92_ImportanceClassType *195*
IRobotSeismicAnalysis_PS_69_DampingType *195*
IRobotSeismicAnalysis_PS_69_SoilType *196*
IRobotSeismicAnalysis_PS_92_StructureType *196*
IRobotSeismicAnalysis_PS_92_SiteType *196*
IRobotSeismicAnalysis_RPA_88_UsageType *197*
IRobotSeismicAnalysis_RPA_88_CategoryType *197*
IRobotSeismicAnalysis_RPA_88_SoilType *198*
IRobotSeismicAnalysis_TURKISH_23098_SoilType *198*
IRobotSeismicAnalysis_UBC_97_SoilType *199*
IRobotSeismicAnalysis_UBC_97_SourceType *199*
IRobotSeismicAnalysis_IBC_2000_SiteClassType *200*
IRobotSeismicAnalysis_EAK_2000_ImportanceFactorType *200*
IRobotSeismicAnalysis_EAK_2000_GroundCategoryType *201*
IRobotSeismicAnalysis_PS_92_SiteEnvelope *201*
IRobotSeismicAnalysis_RPS_2000_ZoneType *202*
IRobotSeismicAnalysis_RPS_2000_SiteType *202*
IRobotSeismicAnalysis_RPS_2000_StructureClass *202*
IRobotSeismicAnalysis_RPS_2000_Params *202*
IRobotSeismicAnalysis_RPA_2003_ZoneType *202*
IRobotSeismicAnalysis_RPA_2003_UsageType *203*
IRobotSeismicAnalysis_RPA_2003_SiteType *203*
IRobotSeismicAnalysis_RPA_2003_Params *203*
IRobotSeismicAnalysis_ITALY_ORDINANZA_Params *203*
IRobotSeismicAnalysis_ITALY_ORDINANZA_SoilType *204*
IRobotSeismicAnalysis_ITALY_ORDINANZA_ZoneType *204*
IRobotSeismicAnalysis_ITALY_ORDINANZA_Spectrum *204*
IRobotSeismicAnalysis_ITALY_ORDINANZA_Direction *204*
IRobotSeismicAnalysis_P_100_2006_Params *204*
IRobotSeismicAnalysis_IBC_2006_Params *205*
IRobotSeismicAnalysis_IBC_2006_SiteClassType *206*
SPECTRAL ANALYSIS PARAMETERS 207
IRobotSpectralAnalysisParams *207*
IRobotSpectralAnalysisSpectrum *207*
IRobotSpectralAnalysisPointsCollection *208*
IRobotSpectralAnalysisAbscissaXAxisType *209*
IRobotSpectralAnalysisOrdinateYAxisType *209*
NON-LINEAR ANALYSIS PARAMETERS 210
IRobotNonlinearAnalysisAlgorithmType *210*
IRobotNonlinearAnalysisParams *210*
BUCKLING ANALYSIS PARAMETERS 212
IRobotBucklingAnalysisParams *212*
IRobotBucklingAnalysisMethod *213*
TIME HISTORY ANALYSIS PARAMETERS 213
IRobotTimeHistoryAnalysisParams *213*
IRobotTimeHistoryAnalysisMethod *215*
IRobotTimeHistoryNewmarkParams *215*
IRobotTimeHistoryModalDecompositionParams *215*

IRobotTimeHistoryPointsCollection *216*
IRobotTimeHistoryFunctionList *217*
IRobotTimeHistoryNonlinearParams *218*
IRobotTimeHistoryPredictorCorectorParams *219*
HARMONIC ANALYSIS PARAMETERS 219
IRobotHarmonicAnalysisParams *219*
PUSH-OVER ANALYSIS PARAMETERS 219
IRobotPushOverAnalysisParams *220*
IRobotPushOverLoadDefinitionMethod *220*
IRobotPushOverDirection *220*
MOVING LOADS 221
IRobotMobileCase *221*
IRobotMobileCaseComponentMngr *223*
IRobotMobileCaseComponent *223*
IRobotMobileCaseFlag *223*
IRobotMobileCaseRoute *224*
IRobotMobileCaseSegmentFactors *225*
IRobotMobileCaseApplicationPlaneType *225*
IRobotVehicleData *226*
IRobotVehicleLoadMngr *226*
IRobotVehicleLoad *227*
IRobotVehicleLoadType *228*
IRobotVehicleDatabaseList *228*
IRobotVehicleDatabase *229*
CODE COMBINATIONS 230
IRobotCodeCombinationEngine *230*
IRobotCodeCombination *230*
IRobotCodeCmbComponentMngr *231*
IRobotCodeCmbFactor *231*
IRobotCodeCmbComponent *231*
IRobotCodeCmbFlag *232*
IRobotCodeCmbRegulations *232*
IRobotCodeCmbActionServer *233*
IRobotCodeCmbActionCoeffType *234*
IRobotCodeCmbCombs *235*
IRobotCodeCmbCombPartType *236*
IRobotCodeCmbCombType *236*
IRobotCodeCmbGenerationParams *237*
IRobotCodeCmbDecidingValueType *238*
IRobotCodeCmbGroup *239*
IRobotCodeCmbOperator *240*
IRobotCodeCmbGroupServer *240*
IRobotCodeCmbGroupRelation *241*
IRobotCodeCmbGroupRelationServer *242*
IRobotCodeCmbActiveCaseInfo *243*
SNOW/WIND LOADS 243
IRobotSnowWindParams *243*
IRobotSnowWindEngine *244*
IRobotSWCodePLParams *244*
IRobotSWCodePLWindZone *246*
IRobotSWCodePLWindSite *247*
IRobotSWCodePLSnowZone *247*
IRobotSWCodeFRParams *248*
IRobotSWCodeFRWindSite *251*
IRobotSWCodeFRWindType *251*
IRobotSWCodeFRSurfaceType *252*

IRobotSWCodeFRSnowType *252*
IRobotSWCodePLWindPressDistribType *252*
IRobotSWCodeECParams *252*
IRobotSWCodeECSiteType *257*
IRobotSWCodeECGroundType *257*
IRobotSWCodeECCdType *258*
GENERATING 3D LOADS 258
IRobotSWStruct3D *258*
IRobotSWStruct3DElement *259*
IRobotSWStruct3DFrame *259*
IRobotSWStruct3DGenParams *259*
IRobotSWStruct3DPurlinGenParams *260*
OBJECTS - COMPLEX ELEMENTS OF THE STRUCTURE 260
IRobotObjObjectServer *261*
IRobotObjObject *263*
IRobotObjPartType *264*
IRobotObjPart *265*
IRobotObjPart2 ..... *265*
IRobotObjPartMain *266*
IRobotObjPartReference *267*
IRobotObjEdge *267*
IRobotObjAttributes *268*
IRobotObjModificationType *269*
IRobotObjOperationType *269*
IRobotObjModification *270*
IRobotObjOperation *270*
IRobotObjOperationCollection *271*
IRobotObjModifExtrusion *271*
IRobotObjModifLathe *271*
IRobotObjModifPyramid *271*
IRobotObjOperTranslation *272*
IRobotObjOperScaling *272*
IRobotObjOperRotation *272*
IRobotObjOperMeshing *273*
IRobotObjModificationCollection *273*
IRobotObjEdgeCollection *273*
IRobotObjLocalXDirDefinitionType *273*
IRobotObjMesh *274*
COMPLEX ATTRIBUTES OF AN OBJECT 275
IRobotThicknessData *275*
IRobotThicknessType *275*
IRobotThicknessUpliftType *275*
IRobotThicknessHomoData *276*
IRobotThicknessHomoType *277*
IRobotThicknessOrthoType *277*
IRobotThicknessOrthoDirType *278*
IRobotThicknessOrthoData *278*
IRobotThicknessMatrix *279*
IRobotThicknessMatrixValue *280*
IRobotSolidPropertiesData *281*
LINEAR RELEASES 281
IRobotLinearReleaseData *282*
IRobotLinearReleaseDefinitionType *282*
IRobotLinearReleaseServer *283*
IRobotLinearReleaseDef *284*
IRobotLinearReleaseDefList *285*

IRobotGroupObjectServer *286*
FINITE ELEMENTS 286
IRobotFiniteElement *286*
IRobotFiniteElementType *287*
IRobotFiniteElementNodes *288*
IRobotFiniteElementServer *289*
MESH GENERATOR 289
IRobotMeshType *290*
IRobotMeshForcingRatio *290*
IRobotMeshImplementDegree *291*
IRobotMeshPanelDivType *291*
IRobotMeshSurfaceFEType *292*
IRobotMeshVolumetricFEType *292*
IRobotMeshDelaunayType *292*
IRobotMeshMethodType *293*
IRobotMeshGenerationType *293*
IRobotMeshMethod *294*
IRobotMeshGeneration *294*
IRobotMeshCoonsParams *295*
IRobotMeshSurfaceFiniteElems *295*
IRobotMeshDelaunayParams *295*
IRobotMeshSurfaceParams *296*
IRobotMeshVolumeParams *297*
IRobotMeshParams *297*
IRobotMeshAccessType *298*
IRobotMeshRefinementType *298*
COMPLEX ATTRIBUTES SHARED BY OBJECTS OF DIFFERENT TYPES 298
MATERIAL 298
IRobotMaterialData *299*
IRobotMaterialType *301*
IRobotMaterialTimberType *302*
IRobotMaterialModel *302*
IRobotMaterialElasticType *303*
IRobotMaterialElasticModel *303*
IRobotMaterialElasticUnloadingMethod *303*
NON-LINEAR LINKS 304
IRobotNonlinearLinkServer *304*
IRobotNonlinearLink *305*
IRobotNonlinearLinkParams *306*
IRobotNonlinearLinkParamsLinear *306*
IRobotNonlinearLinkParamsBLinear *306*
IRobotNonlinearLinkCurveType *306*
IRobotNonlinearLinkParamsParabolic *307*
IRobotNonlinearLinkParamsPlastic *308*
IRobotNonlinearLinkParamsPlasticHardening *308*
IRobotNonlinearLinkParamsGapHook *309*
IRobotNonlinearLinkParamsCustom *309*
IRobotNonlinearLinkParamsCustomSegment *310*
IRobotNonlinearLinkModelType *310*
IRobotNonlinearLinkSemiAxisType *310*
IRobotNonlinearLinkMngr *311*
RESULTS 311
IRobotResultServer *311*
IRobotUniversalResultAccess *312*
IRobotExtremeResultServer *315*
IRobotUniversalResultType *315*

IRobotStructureValues *315*
STRUCTURES CONTAINING COMPLEX CALCULATION RESULTS 316
IRobotReactionData *316*
IRobotDisplacementData *316*
IRobotBarForceData *317*
IRobotBarStressData *317*
IRobotBarDeflectionData *318*
IRobotBarBucklingData *318*
IRobotEigenvalues *319*
IRobotForcesData *319*
IRobotTimeHistoryResults *320*
IRobotExtremeValue *321*
IRobotExtremeParams *322*
IRobotExtremeValueType *322*
CALCULATION RESULTS FOR NODES 327
IRobotNodeResultServer *327*
IRobotReactionServer *328*
IRobotNodeDisplacementServer *330*
IRobotNodeBucklingServer *330*
IRobotPseudostaticForceServer *331*
CALCULATION RESULTS FOR BARS 331
IRobotBarResultServer *331*
IRobotBarForceServer *332*
IRobotBarDeflectionServer *333*
IRobotBarStressServer *333*
IRobotBarBucklingServer *334*
IRobotBarDisplacementServer *334*
DYNAMIC RESULTS 335
IRobotAdvancedResultServer *335*
IRobotEigenvaluesServer *335*
IRobotMassSumServer *336*
IRobotSpectralCoefficients *336*
IRobotEigenvectorsServer *337*
IRobotTimeHistoryResultServer *337*
CALCULATION RESULTS FOR FINITE ELEMENTS 337
IRobotFeLayerType *337*
IRobotFeResultParams *338*
IRobotFeResultDetailed *339*
IRobotFeResultServer *341*
IRobotReinforceCalcMethods *343*
IRobotFeResultPrincipal *343*
IRobotFeResultComplex *345*
IRobotFeResultReinforcement *345*
IRobotFeResultReduced *346*
IRobotFeResultReducedCutPosition *348*
IRobotPanelCutMngr *348*
IRobotPanelCut *349*
IRobotPanelCutDefinitionType *350*
IRobotFeResultType *350*
IRobotFeResultSmoothing *356*
IRobotFeResultKind *356*
IRobotFeExtremeParams *357*
IRobotFeExtremeValue *359*
IRobotFeMultiResultType *360*
IRobotFeMultiExtremeValue *361*
IRobotMaterialQuantitySurvey *361*

IRobotBarSectionQuantitySurvey *362*
IRobotThicknessQuantitySurvey *362*
IRobotStructureQuantitySurvey *363*
EDIT OPERATIONS 364
IRobotTranslateOptions *364*
IRobotStructureEditTools *364*
STOREYS 365
IRobotStorey *365*
IRobotStoreyMngr *366*
CALCULATION MODULE 366
IRobotCalcEngine *366*
IRobotEquationSolvingMethod *368*
IRobotStructureAnalysisParams *368*
IRobotStructureAutoVerificationType *369*
IRobotSparseMSolverParams *369*
IRobotSparseMSolverMethod *370*
IRobotIterativePrediconditionerType *370*
IRobotIterativeSolverParams *371*
IRobotIterativeSolverMethod *371*
IRobotIterativeSolverMemoryUsage *372*
IRobotStructureAnalysisModalParticipationCoeff *372*
IRobotSeismicResultsSaveParams *373*
IRobotSeismicResultsPanelDirection *374*
IRobotModelGenerationParams *374*
IRobotBucklingDeformationParams *375*
DATA PRESENTATION 376
STRUCTURAL AXES 376
IRobotStructuralAxisLabelType *376*
IRobotStructuralAxisSequenceList *376*
IRobotStructuralAxisGridType *377*
IRobotStructuralAxisGrid *377*
IRobotStructuralAxisGridCartesian *378*
IRobotStructuralAxisGridMngr *378*
VIEWS AND LAYOUTS 379
IRobotViewMngr *379*
IRobotLayoutId *380*
GRAPHICAL STRUCTURE VIEWS 384
IRobotViewType *384*
IRobotView *385*
IRobotViewProjection *387*
IRobotViewVisibilityStatusType *388*
IRobotViewVisibilityStatusValue *392*
IRobotViewScaleType *404*
IRobotViewDetailedAnalysis *405*
IRobotViewDetailedAnalysisTableTab *405*
IRobotViewDisplayParams *406*
IRobotViewDisplayAttributes *406*
IRobotViewHiddenLinesDisplayType *410*
IRobotViewDiagramParams *410*
IRobotViewDiagramResultType *411*
IRobotViewDiagramDescriptionType *414*
IRobotViewDiagramFillingType *414*
IRobotViewDiagramSignDifferType *414*
IRobotViewDiagrams *415*
IRobotViewBarMaps *415*
IRobotViewFeMaps *415*

IRobotViewBarMapResultType *415*
IRobotViewBarMapParams *416*
IRobotViewDetailedAnalysisResultType *417*
IRobotViewDetailedAnalysisParams *419*
IRobotViewFeMapResultType *419*
IRobotViewFeMapParams *422*
IRobotViewFeMapLocalSystemType *423*
IRobotViewFeMapCrossPresentationType *424*
IRobotViewFeMapLayerType *424*
IRobotViewFeMapSmoothingType *425*
IRobotView2 ..... *425*
IRobotViewDiagramPositionType *426*
IRobotViewReinforcementResultType *426*
IRobotViewGlobalAnalysisParams *427*
IRobotViewGlobalAnalysisResultsParams *427*
IRobotViewGlobalAnalysisResultsType *428*
IRobotViewGlobalAnalysisParamsType *428*
IRobotViewGlobalAnalysis *430*
IRobotViewScreenCaptureParams *430*
IRobotViewScreenCaptureUpdateType *431*
IRobotView3 ..... *431*
PANEL CUTS 431
IRobotViewPanelCuts *431*
IRobotViewPanelCutParams *431*
INFLUENCE LINES 432
IRobotViewInfluenceLines *432*
IRobotViewInfluenceLinesParams *433*
IRobotViewInfluenceLinesResultType *434*
IRobotViewInfluenceLinesLocalSystemType *439*
IRobotViewInfluenceLinesLayerType *439*
IRobotViewDiagramValueType *440*
TABLES 440
IRobotTableType *440*
IRobotTableDataType *440*
IRobotTableConfigFlag *440*
IRobotTableConfigValue *443*
IRobotTableConfig *443*
IRobotTable *443*
IRobotTableFrame *445*
VIEWERS FOR RTF FILES 445
IRobotRtfView *446*
IRobotHtmlView *446*
SCREEN CAPTURES 447
IRobotScreenCaptureMngr *447*
PRINTOUTS 447
IRobotPrintable *447*
IRobotPrintEngine *448*
IRobotVariableMngr *450*
IRobotVariablePredefinedId *450*
IRobotPageSetup *452*
IRobotOutputFileFormat *455*
IRobotReportTemplateMngr *455*
IRobotReportStdElementRtf *456*
IRobotReportItemList *456*
IRobotReportItem *457*
IRobotPageSetupOrientation *458*

IRobotPageSetupFrameType *458*
IRobotPageSetupTableOfContents *458*
IRobotPageSetupTocLocation *459*
IRobotReportItemType *459*
IRobotExternalPreviewFormat *460*
IRobotVariableMngrExtension *460*
ADD-IN MANAGER 460
IRobotAddIn *460*
IRobotAddInMngr *461*
IRobotCmdList *462*
IRobotCmdInfo *462*
IRobotAddInRegistrar *463*
CONCRETE 463
IRConcrBarSectionData *464*
IRConcrBarSectionGeometryType *464*
IRConcrCalcEngine *464*
PLATE AND SHELL REINFORCEMENT 465
IRConcrReinforceData *465*
IRConcrReinforceDirection *466*
IRConcrConcreteParams *467*
IRConcrSteelParams *468*
IRConcr_PN84_ReinforceData *468*
IRConcr_PN_SteelGrades *470*
IRConcr_PN84_ConcreteGrades *470*
IRConcr_PN84_ExposureRatings *471*
IRConcr_PN99_ExposureRatings *471*
IRConcr_PN99_ReinforceData *472*
IRConcr_PN99_ConcreteGrades *473*
IRConcr_BAEL_ReinforceData *474*
IRConcr_BAEL_CrackingType *476*
IRConcr_BAEL_EnvironmentType *476*
IRConcr_BAEL_WaterLevel *476*
IRConcr_BAEL_ConcreteGrades *477*
IRConcr_BAEL_SteelGrades *477*
IRConcr_PN84_HumidityType *477*
IRConcr_ACI318_ReinforceData *478*
IRConcr_ACI318_LoadActionPeriodType *479*
IRConcr_ACI318_SteelGrades *480*
IRConcr_ACI318_ConcreteParams *480*
IRConcr_ACI318_BarDim *481*
IRConcr_ACI318_MetricBarDIm *482*
IRConcrReinforceBarType *483*
IRConcr_BS8110_ReinforceData *483*
IRConcr_BS8110_ExposureRatings *484*
IRConcr_BS8110_ConcreteAge *485*
IRConcr_BS8110_ConcreteGrades *486*
IRConcr_BS8110_PartialSafetyFactors *486*
IRConcrSlabRequiredReinfCalcParams *487*
IRConcrSlabRequiredReinfEngine *488*
IRConcrReinforceCalcType *488*
IRConcrReinforceData2 ..... *488*
IRConcrReinforceDataMain *488*
IRConcr_SNIP_ReinforceData *489*
IRConcr_SNIP_SteelGrades *489*
IRConcr_SNIP_ConcreteGrades *489*
IRConcr_SNIP_ConcreteTypes *490*

IRConcr_SNIP_CuringMethods *490*
IRConcr_SNIP_ConcreteParams *490*
IRConcr_SNIP_Exposure *490*
IRConcr_EC2_ConcreteGrades *490*
IRConcr_EC2_ITALIAN_SteelGrades *491*
IRConcr_EC2_ExposureRatings *491*
IRConcr_EC2_ReinforceData *491*
IRConcr_EC2_NAD *492*
IRConcrMemberRequiredReinfEngine *492*
IRConcrMemberRequiredReinfCalcParams *492*
IRConcrBeamCalcPointDefinitionType *493*
THEORETICAL REINFORCEMENT FOR CONCRETE BARS 494
IRBestCodeService *494*
IRBestCalcParamsDlg *494*
IRBestMemberDlg *494*
IRBestParamSet *495*
IRBestCalcParamsData *496*
IRBestCalcParamsDataDoubleValue *496*
IRBestCalcParamsDataIntegerValue *498*
IRBestCalcParamsDataStringValue *499*
IRBestCalcParamsDataList *499*
IRBestMemberData *500*
IRBestMemberDataDoubleValue *500*
IRBestMemberDataIntegerValue *501*
IRBestResults *501*
IRBestResultsDoubleValue *501*
IRBestResultsIntegerValue *503*
IRBestCalcErrors *503*
IRBestCalcWarnings *504*
IRBestResultsStringValue *505*
IRBestForceData *505*
IRBestDimParams *505*
IRBestDimParamsIntegerValue *506*
IRBestCodeCalcEngine *506*
IRBestMemberDataStringValue *507*
IRBestMemberType *507*
IRBestPlateCalcParamsDlg *508*
IRBestCodeCalculationType *508*
IRBestCodeServiceExt *508*
IRBestCalculationType *508*
IRBestDimParamsDoubleValue *509*
IRBestBendType *509*
IRBestDirection *509*
IRBestLevel *509*
IRBestForceDataDoubleValue *509*
IRBestCoordSystem *510*
STEEL AND TIMBER DIMENSIONING 510
IRDimStreamType *510*
IRDimStream *510*
IRDimMembDefType *511*
IRDimMembDefMatType *511*
IRDimMembDefLengthDataType *511*
IRDimMembDefBucklingDataType *511*
IRDimMembDefDispDataType *511*
IRDimMembDefDeflDataType *512*
IRDimMembDef *512*

IRDimEffDefParamType *512*
IRDimEffDefDirType *512*
IRDimEffDefIntPsType *513*
IRDimEffDef *513*
IRDimMatDefType *514*
IRDimMatDefValType *514*
IRDimMatDefLongExValType *515*
IRDimMatDefDblExValType *515*
IRDimMatDef *515*
IRDimProfDefType *516*
IRDimProfDefItemType *517*
IRDimProfDefValType *517*
IRDimProfDef *518*
IRDimCalcStateFlagType *518*
IRDimCalcStateParamType *518*
IRDimCalcStateParamValue *518*
IRDimCalcStateValueType *519*
IRDimCalcState *519*
IRDimMembSrv *519*
IRDimMembCalcRetValue *519*
IRDimMembCalcBuckType *519*
IRDimMembResTableLineType *520*
IRDimMembResTableComp *520*
IRDimMembRes *520*
IRDimMembCalc *521*
IRDimCodeService *522*
IRDimClient *522*
IRDimUnitType *522*
IRDimUnits *522*
IRDimServer *523*
CODE EC3 523
IRDimMembParamsEC3 ..... *523*
IRDimLoadTypeEC3 ..... *525*
IRDimLaterBuckTypeEC3 ..... *525*
IRDimLoadLevelEC3 ..... *526*
IRDimLatBuckCoeffDiagramEC3 ..... *526*
IRDimBuckDiagramEC3 ..... *527*
IRDimCodeResEC3 ..... *528*
IRDimYieldStrengthTypeEC3 ..... *536*
CODE CB71 537
IRDimMembParamsCB71 ..... *537*
IRDimCodeResCB71 ..... *539*
IRDimBuckDiagramCB71 ..... *545*
IRDimLaterBuckTypeCB71 ..... *545*
IRDimLoadLevelCB71 ..... *546*
IRDimLoadTypeCB71 ..... *546*
IRDimFireMemberPositionsCB71 ..... *547*
IRDimFireProtectionTimesCB71 ..... *547*
CONNECTION MODULE 547
IRJointConnectionDefType *548*
IRJointConnectionType *548*
IRJointConnectionInfo *549*
IRJointWebFlangeRelativePos *549*
IRJointConnection *549*
IRJointWeld *550*
IRJointPlate *550*

IRJointBolts *551*
IRJointBoltType *551*
IRJointWebType *551*
IRJointConnectionServer *552*
IRJointProfile *553*
IRJointLoad *553*
IRJointLoadType *553*
KNEE CONNECTION 553
IRJointKneeBolts *554*
IRJointKneeBracket *554*
IRJointKneeWebPlate *554*
IRJointKneeType *555*
IRJointKneeFixType *555*
IRJointKneeReinfType *555*
IRJointKnee *556*
IRJointKneeWebStiffType *557*
IRJointKneeDiagonalStiffType *557*
IRJointKneeDiagonalStiff *557*
IRJointKneeStiffColumn *558*
IRJointKneeLoad *558*
FOOTING CONNECTIONS 558
IRJointAnchorBolt *558*
IRJointAnchorPlate *559*
IRJointAnchorType *559*
IRJointAnchor *559*
IRJointFootBolts *560*
IRJointFootPlateType *560*
IRJointFootPlate *561*
IRJointWedge *561*
IRJointWedgeType *561*
IRJointBearingPlate *562*
IRJointFootWelds *562*
IRJointColumnBracket *563*
IRJointAnchorPlateType *563*
IRJointFootStiffenerVert *563*
IRJointFootStiffenerHoriz *564*
IRJointFootMaterials *564*
IRJointColumnSquare *564*
IRJointFootStiffenerSimple *565*
IRJointFootStiffenerComplex *565*
IRJointFootStiffType *565*
PINNED COLUMN BASE 566
IRJointPinnedColumnBaseStiffType *566*
IRJointPinnedColumnBase *566*
IRJointPinnedLoad *568*
FIXED SPREAD FOOTING 568
IRJointFixedLoad *568*
IRJointFixedColumnBase *569*
IRJointFixedFootWelds *570*
IRJointFixedColumnBaseStiffType *570*
SPREAD FOOTING 570
IRJointConcreteMaterials *570*
IRJointConcreteColumn *571*
IRJointSpreadFootingType *571*
IRJointConcreteColumnFoundation *572*
IRJointColumnBasePlateCalcModel *573*

IRJointConcreteColumnLoad *573*
ANGLE CONNECTION 574
IRJointAngleBolts *574*
IRJointAngle *574*
IRJointAngleType *575*
IRJointBeamCut *575*
IRJointWithAngles *575*
IRJointAngleProfilePosition *576*
IRJointAngleLoad *576*
TUBE CONNECTION 577
IRJointTube *577*
IRJointTubeType *577*
IRJointTubeProfileType *578*
IRJointTubeFlangeProfile *578*
IRJointTubeDiagProfile *579*
IRJointTubePostProfile *579*
IRJointTubeLoad *580*
GUSSET PLATE 581
IRJointGussetBoltsDiag *581*
IRJointGussetWeldsDiag *581*
IRJointGussetDiagonale *581*
IRJointGussetFixType *582*
IRJointGussetSimpleProfilePosition *582*
IRJointGussetSimple *583*
IRJointGussetSimplePlate *583*
IRJointGussetCornersType *584*
IRJointGussetSimpleAttachment *584*
IRJointGussetSimpleAttachBolts *584*
IRJointGussetSimpleAttachBoltsHorizontal *585*
IRJointGussetSimpleAttachBoltsVertical *585*
IRJointGussetSimpleAttachWelds *585*
IRJointGussetCrossProfileCutting *585*
IRJointGussetCrossPlate *586*
IRJointGussetCross *587*
IRJointGussetFlangeProfileCutting *587*
IRJointGussetFlangePlate *587*
IRJointGussetFlange *588*
IRJointGussetSimpleLoad *589*
IRJointGussetCrossLoad *589*
IRJointGussetFlangeLoad *590*
IRJointGussetDiagonalePositionType *590*
IRJointGussetFlangePlateRegularType *591*
IRJointBeamGirderBolts *591*
IRJointBeamGirderStiffener *592*
IRJointBeamGirderPlate *593*
IRJointBeamGirderSeat *594*
IRJointBeamGirder *594*
IRJointBeamGirderLoad *596*
IRJointBeamGirderSeatBolts *597*
IRJointConnectorsType *597*
IRJointElementType *598*
IRJointBeamGirderAngle *599*
IRJointBeamGirderBeam *600*
CORRECTION OF THE STRUCTURE 600
IRobotBarIntersectRelationship *600*
IRobotStructureGeoAnalyser *601*

STR FILES ANALYSER 602
IRobotSTRFileAnalyser *602*
ROBOT KERNEL 603
IRobotKernel *603*
IRobotKernelPreferences *604*

Robot Object Model

When modeling a structure, one uses a certain set of concepts that facilitate the process of defining and analyzing the structure. Here belong node, bar, load case and many others. While working in Robot, one encounters such concepts as project – represented by the RTD file, graphical viewers and tables – presenting different aspects of data describing a structure, as well as many other abstractions. In order to make it possible to use the above-mentioned concepts at the level of a programming language, they have been formalized and defined according to COM standards.
Program members and data types used to define structures are described in Robot Object Model by means of appropriate interfaces. An interface is understood to mean a set of data, mutually linked in a logical manner, and a set of operations that can be performed on the data. The data are called attributes or members, and operations that can be performed are called functions or methods of the interface. The set of interface functions defines its functionality. Each interface has got a unique name. For instance, a node of a structure is represented by means of the IRobotNode interface. Among the data of the interface one finds, among others, three real numbers corresponding to node coordinates. In order to take advantage of the functionality of a given interface, one should create an object that realizes the interface. The manner of creating an object depends on the currently used programming language. In Visual Basic, for instance, a new object is created by means of the New operator or CreateObject function.
Additionally, there is a certain interface type that describes the set of constant values or identifiers. Such interface represents enumeration type. The IRobotProjectType interface is a good example here. Identifiers referring to different types of projects available in Robot are its members. In order to use a member of such an interface, it suffices to provide its name. There is no necessity to create an object to realize it.
Robot Object Model is a set of such interfaces. It is contained in the robotom.tlb file in the standard binary format, so that interfaces available in the model could be recognized by the compilers of programming languages and by the system software. The TLB standard extension comes from the Type Library name, as the interface is often, especially at the level of a programming language, identified with data type. Therefore, one can refer to an object realizing the interface called InterfaceName by calling it an object of the InterfaceName type.

Naming conventions

All names begin with a capital letter. If a name consists of more than one word, each name element begins with a capital letter.
Additionally, an interface name begins with the IRobot prefix, but for interfaces associated with a specialized module the prefix coresponds to the module name (eg IRJoint for interfaces defining the connection module of Robot). The convention has been adopted to minimize the possibility of conflicting names in programming languages that cannot use many independent name spaces.
The names of members of enumeration interface consist, in turn, of capital letters, and particular segments are separated by the underline sign ("_"). Additionally, each name begins with I_FLIN_, where FLIN refers to the first letters of interface name, with the prefix omitted. For instance, names of all elements of IRobotProjectType interface begin with the I_PT_ prefix (e.g. I_PT_FRAME_2D). Such convention facilitates the creation of unique names. The uniqueness of names of members of

enumeration interfaces is necessary, for they are often interpreted in programming languages as global constants that should have unique names.

Units

All functions of Robot Object Model use the standard SI units, independently of the current settings in Robot options.

Global data types

IRobotValuesArray

Interface representing a one-dimensional table of real numbers

Properties:

Count : long

number of elements in the table (at the same time, it is the index of the last element in the table)

Operations:

SetSize (count : long)

The function sets the table size. Table elements are indexed from 1 to table size.

Set (idx : long, val : double)

The function sets the value of table element with the given index. Table elements are indexed from 1 to Count.

Get (idx : long) : double

The function returns the value of table element with the given index. Table elements are indexed from 1 to Count.

IRobotNamesArray

The interface describing a one-dimensional table of names represented by character strings.

Properties:

Count : long

number of elements in the table (it is also the index of the last element in the table)

Operations:

SetSize (count : long)

The function sets a new table size. Table elements are indexed from 1 to Count.

Get (idx : long) : string

The function returns table element with the given index. Table element are indexed from 1 to Count.

Set (idx : long, text : string)
The function sets the value of the given table element. The table elements are indexed from 1 to Count.

Find (_to_find : string, _start_idx : long) : long
Function returns index of the name specified in the table. Table elements are searched beginning with the determined start index. If the specified name is not found, function returns value -1.
Available since version 3.5.

IRobotNumbersArray

The interface representing a table of integers. Table elements are indexed from 1 to Count.

Properties:

Count : long
Table size (at the same time, this is the index of the last element in the table)

Operations:

SetSize (count : long)
The function sets a new table size. Table elements will be indexed from 1 to table size.

Get (idx : long) : long
The function returns a table element with the indicated index. Table elements are indexed from 1 to Count.

Set (idx : long, val : long)
The function sets the value of table element with the given index. Table elements are indexed from 1 to Count.

IRobotObjectsArray

The interface providing access to the table of objects

Properties:

Count : long
Number of objects in the table

Operations:

SetSize (count : long)

The function sets the table size. Table elements are indexed from 1 to table size. The table size is also the number of its elements.

Set (idx : long, object : IDispatch)
The function sets the indicated table element.

Get (idx : long) : IDispatch
The function returns the indicated table element.

IRobotNumbersCollection

Collection of numbers (integers)

Properties:

Count : long
number of elements in the collection

Operations:

Get (idx : long) : long
The function returns the required collection element. The elements are indexed from 1 to Count.

IRobotCodeRegistrar

In order to simplify the process of registering external codes extending the Robot program, RobotCodeRegistrar has been defined.

Properties:

CodeType : IRobotCodeType
code type

CodeName : string
code name

Manufacturer : string
name of the producer/dealer of the given code implementation

Guid : string
the text containing CLSID of the main external code implementation component (the field may be filled alternatively from ProgId).

ProgId : string
the text containing ProgID of the main external code implementation component (the field may be set alternatively with the Guid field)

Operations:

Register ()
The function registers code implementation on the basis of the set values.

Unregister ()
The function unregisters the code implementation on the basis of the set values.

IRobotWindow

General interface describing a window in the Robot program. It describes both the main application window and windows which comprise individual views.

Properties:

State : IRobotWindowState
state of the window
Available since version 1.7.

IsActive : bool
indicator of window activity
Available since version 1.7.

Handle : long
system window handle
Available since version 1.7.

Caption : string

Available since version 3.

Operations:

Activate ()
Function results in activating the window.
Available since version 1.7.

SendMessage (_msg_id : long, _w_param : long, _l_param : long)
The function allows to send a message to the window.
Available since version 3.

IRobotWindowState

Available window states.

Properties:

I_WS_NORMAL : = 0
Available since version 1.7.

I_WS_MAXIMIZE : = 1
Available since version 1.7.

I_WS_MINIMIZE : = 2
Available since version 1.7.

IRobotProtectionInfo

Object provides access to information about protection of the Robot program and its extensions.

Operations:

IsExtensionEnabled (_ext_num : long) : bool
Function returns value different from zero (True), if the external module (application) - Robot program extension - has been activated in the Robot program protection system.
Available since version 2.0.

IRobotComponentFactory

Auxiliary component factory. It should be applied instead of the standard New operator for creation of "free" components in case when the application is to cooperate with both the Robot program and Kernel, depending on the user selection.
Available since version 2.5.

Operations:

Create (_cmpnt_type : IRobotComponentType) : IDispatch
Function creates a new component of the indicated type.
Available since version 2.5.

CreateExt (_extmdl_name : string, _cmpnt_type : long) : IDispatch
Function creates a new component supported by the indicated extension of the Robot program kernel.
Available since version 2.5.

IRobotComponentType

Available component types that can be created by means of the component factory.
Available since version 2.5.

Properties:

I_CT_VALUES_ARRAY : = 1

Available since version 2.5.

I_CT_NUMBERS_ARRAY : = 2

Available since version 2.5.

I_CT_NAMES_ARRAY : = 3

Available since version 2.5.

I_CT_OBJECTS_ARRAY : = 4

Available since version 2.5.

I_CT_GEO_POINT_2D : = 5

Available since version 2.5.

I_CT_GEO_POINT_3D : = 6

Available since version 2.5.

I_CT_GEO_CURVE_DIV : = 7

Available since version 2.5.

I_CT_GEO_SEGMENT : = 8

Available since version 2.5.

I_CT_GEO_SEGMENT_LINE : = 9

Available since version 2.5.

I_CT_GEO_SEGMENT_ARC : = 10

Available since version 2.5.

I_CT_GEO_OBJECT : = 11

Available since version 2.5.

I_CT_GEO_POLYLINE : = 12

Available since version 2.5.

I_CT_GEO_CONTOUR : = 13

Available since version 2.5.

I_CT_GEO_ARC : = 14

Available since version 2.5.

I_CT_GEO_CIRCLE : = 15

Available since version 2.5.

I_CT_GEO_SEGMENT_COLLECTION : = 16

Available since version 2.5.

I_CT_GEO_POINT_3D_COLLECTION : = 17

Available since version 2.5.

I_CT_GEO_LAYER : = 18

Available since version 2.5.

I_CT_EMITTER : = 19

Available since version 2.5.

I_CT_JOINT_LOAD : = 201

Available since version 2.5.

I_CT_JOINT_KNEE_LOAD : = 202

Available since version 2.5.

I_CT_JOINT_ANGLE_LOAD : = 203

Available since version 2.5.

I_CT_JOINT_TUBE_LOAD : = 204

Available since version 2.5.

I_CT_JOINT_PINNED_LOAD : = 205

Available since version 2.5.

I_CT_JOINT_FIXED_LOAD : = 206

Available since version 2.5.

I_CT_JOINT_GUSSET_SIMPLE_LOAD : = 207

Available since version 2.5.

I_CT_JOINT_GUSSET_CROSS_LOAD : = 208

Available since version 2.5.

I_CT_JOINT_GUSSET_FLANGE_LOAD : = 209

Available since version 2.5.

I_CT_MODIF_EXTRUSION : = 29

Available since version 2.5.

I_CT_MODIF_LATHE : = 30

Available since version 2.5.

I_CT_MODIF_PYRAMID : = 31

Available since version 2.5.

I_CT_OPER_TRANSLATION : = 32

Available since version 2.5.

I_CT_OPER_SCALING : = 33

Available since version 2.5.

I_CT_OPER_ROTATION : = 34

Available since version 2.5.

I_CT_OPER_MESHING : = 35

Available since version 2.5.

I_CT_SPECTRAL_ANALYSIS_POINTS_COLLECTION : = 36

Available since version 2.5.

I_CT_SPECTRAL_ANALYSIS_SPECTRUM : = 37

Available since version 2.5.

I_CT_CASE_ANALYSIS_MODES_FILTER : = 38

Available since version 2.5.

I_CT_FE_RESULT_PARAMS : = 39
object steering the parameters of getting results for finite elements
Available since version 2.5.

I_CT_RTF_VIEW : = 40
view for RTF files
Available since version 2.5.

I_CT_POINTS_ARRAY : = 41
table of points in the 3D space (IRobotPointsArray)
Available since version 3.

I_CT_EXTREME_PARAMS : = 42
parameters of taking extreme values
Available since version 3.5.

I_CT_STRUCTURE_GEO_ANALYSER : = 43
structure geometry analyzer
Available since version 3.5.

I_CT_FE_EXTREME_PARAMS : = 44

Available since version 4.

I_CT_FE_MULTI_RESULT_TYPE : = 45

Available since version 4.1.

I_CT_SW_STRUCT3D_GEN_PARAMS : = 46
parameters of generation of a 3D structure model for snow/wind loads (IRobotSWStruct3DGenParams object type)

I_CT_SW_STRUCT3D_PURLIN_GEN_PARAMS : = 47
IRobotSWStruct3DPurlinGenParams object type

I_CT_SW_STRUCT3D : = 48
3D structure model for generation of snow/wind loads (IRobotSWStruct3D object type)

I_CT_SW_STRUCT3D_FRAME : = 49
IRobotSWStruct3DFrame object type

I_CT_SW_STRUCT3D_ELEMENT : = 50
IRobotSWStruct3DFrame object type

I_CT_VIEW_SCREEN_CAPTURE_PARAMS : = 51
I_CT_HTML_VIEW : = 52
view for the HTML format files
Available since version 7.5.

I_CT_STRUCTURE_MERGE_DATA : = 53

Available since version 8.

I_CT_NUMBERS_DICTIONARY : = 54

Available since version 8.1.

IRobotDegreeOfFreedom

Available degrees of freedom.
Available since version 2.5.

Properties:

I_DOF_UX : = 0

Available since version 2.5.

I_DOF_UY : = 1

Available since version 2.5.

I_DOF_UZ : = 2

Available since version 2.5.

I_DOF_RX : = 3

Available since version 2.5.

I_DOF_RY : = 4

Available since version 2.5.

I_DOF_RZ : = 5

Available since version 2.5.

IRobotPointsArray

Table of points in the 3D space.
Available since version 3.

Properties:

Count : long
number of elements (at the same time, it is the index of the last table element)
Available since version 3.

Operations:

SetSize (_size : long)
Function sets a table size.

Available since version 3.

Set (_idx : long, _x : double, _y : double, _z : double)
Function saves the specified point coordinates under the indicated index to the table. Table elements are indexed from 1 to Count.
Available since version 3.

Get (_idx : long, _x : double*, _y : double*, _z : double*)
Function takes point coordinates saved in the table under the indicated index. Table elements are indexed from 1 to Count.

Available since version 3.

IRobotSerializable

Interface of the object whose data can be saved in/read from the stream.
Available since version 8.

Operations:

Write (_dest_stream : IUnknown) : bool
Function saves object data in the specified stream.
Available since version 8.

Read (_source_stream : IUnknown) : bool
Function reads object data from the specified stream.
Available since version 8.

Geometric data types

IRobotGeoPoint2D

The interface describing a point on a plane

Properties:

X : double
x coordinate

Y : double
Y coordinate

Operations:

Get (x : double*, y : double*)
The function takes all point coordinates.

Set (x : double, y : double)

The function sets all point coordinates.

IRobotGeoPoint3D

The interface describing a point in space.

Properties:

X : double
x coordinate

Y : double
y coordinate

Z : double
z coordinate

Operations:

Get (x : double*, y : double*, z : double*)
The function gives back all point coordinates.

Set (x : double, y : double, z : double)
The function sets all point coordinates.

IRobotGeoCoordinateAxis

Coordinate axes.

Properties:

I_GCA_OX : = 1
axis OX

I_GCA_OY : = 2
axis OY

I_GCA_OZ : = 3
axis OZ

I_GCA_NONE : = 0
no coordinate axis is indicated
Available since version 7.

IRobotGeoCoordinateSystem

Types of coordinate systems

Properties:

I_GCS_GLOBAL : = 0
in global system

I_GCS_LOCAL : = 1
in local system

I_GCS_LOCAL_AFTER_OFFSET : = 1
local coordinate system of a bar in the position considering translation

I_GCS_LOCAL_ORIGINAL : = 2
local coordinate system of a bar

IRobotGeoSegmentType

Types of segments that may compose complex geometrical objects.

Properties:

I_GST_NONE : = 0
undefined

I_GST_LINE : = 1
line (with a rounding)

I_GST_ARC : = 2
arc

IRobotGeoObjectType

Types of geometrical objects.

Properties:

I_GOT_NONE : = 0
undefined

I_GOT_POLYLINE : = 1
polyline

I_GOT_CONTOUR : = 2
contour

I_GOT_ARC : = 4
arc

I_GOT_CIRCLE : = 7

circle

I_GOT_INTERSECTION : = 100
overlap of objects

IRobotGeoCurveDiv

The data type describing object discretization.

Properties:

Mode : long
0 - fixed number of divisions
1 - number of divisions per 360 degrees

N : long
number of divisions

IRobotGeoSegment

Auxiliary interface for the definition of polylines and contours whose fragments may take the form of arcs or roundings.

Properties:

Type : IRobotGeoSegmentType
segment type

P1 : IRobotGeoPoint3D
initial point of a segment (end point is taken from the next segment)

IRobotGeoSegmentLine

Derived from IRobotGeoSegment
Auxiliary interface for defining polylines and contours - a segment with or without a rounding.

Properties:

NDiv : long
number of divisions - discretization of the segment

Round : bool
rounded or not

Radius : double
rounding radius

Div : IRobotGeoCurveDiv
rounding discretization

IRobotGeoSegmentArc

Derived from IRobotGeoSegment
Auxiliary interface for polyline and contour definition - arc.

Properties:

P2 : IRobotGeoPoint3D
point on an arc

Div : IRobotGeoCurveDiv
arc discretization

Operations:

Set (p1 : IRobotGeoPoint3D, p2 : IRobotGeoPoint3D)
The function stets the initial arc point P1 and a point along the arc P2

IRobotGeoObject

The basic interface describing a geometrical object

Properties:

Type : IRobotGeoObjectType
object type

Operations:

Initialize ()
The function initializes internal data on the basis of the data defining the object.

IRobotGeoPolyline

Derived from IRobotGeoObject
Interface describing a polyline

Properties:

Segments : IRobotGeoSegmentCollection
collection of segments describing a polyline

Operations:

Clear ()

The function clears a collection of segments of a polyline.

Add (segment : IRobotGeoSegment)
The function adds a new segment to the polyline.

IRobotGeoSegmentCollection

Derived from IRobotCollection
Collection of segments of geometrical objects.

IRobotGeoContour

Derived from IRobotGeoObject
Interface describing a contour.

Properties:

Segments : IRobotGeoSegmentCollection
collection of segments describing a contour (the contour is closed automatically; one does not have to define the last segment)

Operations:

Clear ()
The function clears a segment collection

Add (segment : IRobotGeoSegment)
The function adds a new segment. The segment may be a line or an arc.

IRobotGeoArc

Derived from IRobotGeoObject
Interface describing an arc

Properties:

P1 : IRobotGeoPoint3D
initial point of the arc

P2 : IRobotGeoPoint3D
final point of the arc

P3 : IRobotGeoPoint3D
a point along the arc

IRobotGeoCircle

Derived from IRobotGeoObject

Interface describing a circle.

Properties:

P1 : IRobotGeoPoint3D
A point belonging to a circle

P2 : IRobotGeoPoint3D
A point belonging to a circle

P3 : IRobotGeoPoint3D
A point belonging to a circle

PC : IRobotGeoPoint3D
The circle center

IRobotGeoPoint3DCollection

Derived from IRobotCollection
Collection of points.

IRobotGeoLayer

Definition of a layer as a space sector. Layer is defined by specifying a cutting plane (by means of three points: P1, P2 and P3) and the fourth PDir point determining a half-space direction. Additionally, layer thickness may be determined by specifying the Thickness value. If the layer thickness is not specified, the layer describes the entire half-space.

Properties:

P1 : IRobotGeoPoint3D
first point (P1) defining the cutting plane
Available since version 2.0.

P2 : IRobotGeoPoint3D
second point (P2) defining the cutting plane
Available since version 2.0.

P3 : IRobotGeoPoint3D
third point (P3) defining the cutting plane
Available since version 2.0.

PDir : IRobotGeoPoint3D
point defining direction (i.e. it indicates one of the half-spaces determined by the cutting plane)
Available since version 2.0.

Thickness : double

layer thickness
Available since version 2.0.

IsThickDefined : bool
auxiliary flag indicating if the layer thickness has been defined
Available since version 2.0.

IRobotGeoArcDefinitionMethod

Available methods of arc definition.
Available since version 3.

Properties:

I_GADM_CENTER_BEGIN_END : = 0

Available since version 3.

I_GADM_BEGIN_MIDDLE_END : = 1

Available since version 3.

IRobotGeoCoordinateAxisSense

Senses of the coordinate system axes.
Available since version 4.5.

Properties:

I_GCAS_OX_PLUS : = 1

Available since version 4.5.

I_GCAS_OY_PLUS : = 2

Available since version 4.5.

I_GCAS_OZ_PLUS : = 3

Available since version 4.5.

I_GCAS_OX_MINUS : = -1

Available since version 4.5.

I_GCAS_OY_MINUS : = -2

Available since version 4.5.

I_GCAS_OZ_MINUS : = -3

Available since version 4.5.

IRobotNumbersDictionary

Dictionary of pairs (integer, integer).
Available since version 8.1.

Properties:

Count : long
Number of pairs defined in the dictionary
Available since version 8.1.

Operations:

Set (_key : long, _value : long)
Function adds a new pair to the dictionary. If a pair with the specified key already exists in the dictionary, its value will be replaced.
Available since version 8.1.

Find (_key : long) : long
Function returns an index of the pair with the specified key. If the pair is not found, function returns zero value.
Available since version 8.1.

Delete (_key : long)
Function deletes a pair with the specified key.
Available since version 8.1.

Get (_idx : long, _key : long*, _value : long*)
Function returns a pair of the specified index.
Available since version 8.1.

FindGet (_key : long, _value : long*) : long
Function returns an index of the pair with the specified key and a value assigned to the key. If the pair is not found, function returns zero value.
Available since version 8.1.

Application - main object of the model

IRobotApplication

Interface representing the entire application is the main object of the Robot Object Model.

Properties:

Visible : bool
flag of the main Robot window visibility

Interactive : bool
flag indicating if interactive work with the program is expected

Project : IRobotProject
current project

Version : double
indicator of Robot Object Model saved in the following format [Version].[Extended] (e.g. 1.02)

Preferences : IRobotPreferences
The object describing application parameters.

Window : IRobotWindow
main window of Robot program
Available since version 1.7.

CmpntFactory : IRobotComponentFactory

Available since version 2.5.

Kernel : IRobotKernel
component enabling access to the object of the Robot application by means of the IRobotKernel interface
Available since version 2.5.

ProgramVersion : string
designation of the Robot program version
Available since version 3.

UserControl : bool
flag used for giving/taking back program navigation to/from the interactive user

Available since version 3.

Operations:

Quit (quit_option : IRobotQuitOption)
Function closes the program.
Available since version 1.7.

IRobotPreferences

Interface describing the current settings for the entire application.

Properties:

OpenGL : bool
flag that enables switching on/off OpenGL graphics
Available since version 5.5.

Operations:

GetDirectory (robot_dir : IRobotDirectory) : string
The function returns full path to the desired Robot folder.

GetLanguage (lang_id : IRobotLanguage) : int
Function returns the current setting of the specified language. Code (number) of the appropriate country is returned (e.g. 33 for France).

IRobotDirectory

The set of identifiers representing folders used by Robot program.

Properties:

I_D_EXE : = 1
folder with executable files

I_D_RES : = 2
folder with resources

I_D_CONF : = 3
folder with configuration files

I_D_USER_CONF : = 4
folder with user's configuration files

I_D_TEMPLATE : = 5
folder with template files

I_D_HELP : = 6
folder with help files

I_D_USER_OUTPUT : = 7
user's target (start-up) folder

I_D_MAIN : = 8
the main Robot folder

I_D_USER_MAIN : = 9

main user folder

I_D_USER_PROJECTS : = 10
project folder

I_D_USER_TEMPLATE : = 11

Available since version 3.

IRobotLanguage

Properties:

I_L_WORK : = 1
interface language

I_L_PRINTOUT : = 2
printout language

I_L_REGIONAL : = 3
regional language

IRobotQuitOption

Set of identifiers describing the manner of exiting the program.

Properties:

I_QO_DISCARD_CHANGES : = 0
changes made in the project will not be saved
Available since version 1.7.

I_QO_PROMPT_TO_SAVE_CHANGES : = 1
if changes that require saving to file have been made in the project, the program will display a message asking if the changes are to be saved
Available since version 1.7.

I_QO_SAVE_CHANGES : = 2
changes will be saved automatically
Available since version 1.7.

IRobotDirectoryExtension

Set of identifiers representing extensions for Robot program folders.
Available since version 3.5.

Properties:

I_DE_TEMPLATE : = 51
extension for the TEMPLATE folder
Available since version 3.5.

I_DE_USER_CONF : = 52

Available since version 3.5.

I_DE_CONF : = 53

Available since version 3.5.

I_DE_USER_TEMPLATE : = 54

Available since version 3.5.

IRobotPreferencesEvents

Events related to the change of settings for the whole application.

Operations:

OnDialogOK ()
Event generated after accepting changes made in the program settings dialog box by the user.

Project

Robot Object Model allows one to manage an entire project (task) saved to an RTD-format file.

IRobotProjectType

A set of identifiers is defined to refer to structure types recognized by Robot.

Properties:

I_PT_FRAME_2D : int = 1
I_PT_TRUSS_2D : int = 2
I_PT_GRILLAGE : = 3
I_PT_FRAME_3D : = 4
I_PT_TRUSS_3D : = 5
I_PT_PLATE : = 6
I_PT_SHELL : = 7
I_PT_AXISYMMETRIC : = 8
I_PT_VOLUMETRIC : = 9

I_PT_CONCRETE_BEAM : = 10
I_PT_CONCRETE_COLUMN : = 11
I_PT_FOUNDATION : = 12
I_PT_PARAMETRIZED : = 13
I_PT_STEEL_CONNECTION : = 14
I_PT_SECTION : = 15
I_PT_PLANE_STRESS : = 16
I_PT_PLANE_DEFORMATION : = 17
I_PT_CONCRETE_DEEP_BEAM : = 18
deep beam
Available since version 2.5.

IRobotProject

RobotProject interface allows for managing all the data saved in an RTD-format file.

Properties:

ComponentMngr : IRobotProjectComponentMngr
manager of all project components

ViewMngr : IRobotViewMngr
manager controlling all views related to the project

PrintEngine : IRobotPrintEngine
module that prints project documentation

Structure : IRobotStructure
the object representing a structure defined in the project

Name : string
project name

FileName : string
path to the file where the project is saved

FileInsertParams : IRobotFileInsertParams
collection of parameter related to the manner of inserting into the project the currently opened external file

ExtFileParams : IDispatch

set of parameters opened by means of ReadExtFileParams() function from an external-format file

CalcEngine : IRobotCalcEngine
object representing the calculation module of Robot

Type : IRobotProjectType
project type (type of structure defined in the project)

IsActive : bool
A flag indicating if the project is active or not - opened from a file or defined by the New() function. It can happen that in the Robot program there is no project opened. In such a situation the IsActive flag equals zero (False).

Connections : IRJointConnectionServer
Server of connections defined in the project.

ExtFileName : string
the path to the last external file (read by means of the OpenExtFile function) in a non-RTD format.

Preferences : IRobotProjectPreferences
the object describing job parameters

UniqueId : string
unique string of characters enabling project identification (RTD file) independently of the file name and its location on disk, which may change while working with the project;
(available since version 1.7)

DimServer : IRDimServer
design and verification server for steel and timber members
Available since version 2.5.

ConcrReinfEngine : RConcrCalcEngine
calculation module for the concrete reinforcement
Available since version 5.5.

AxisMngr : IRobotStructuralAxisGridMngr

Operations:

New (prj_type : IRobotProjectType)
The function creates a new project of the indicated type.

Open (file_path : string)
The function opens an existing project from the indicated file in RTD format.

Save ()

The function saves the current project to the file whose full access path is described by the FileName variable.

SaveAs (file_path : string)

The function saves the current project to the file accessed by the indicated path.

Close ()

The function closes the current project without saving changes.

OpenExtFile (file_path : string, format : IRobotExternalFileFormat, ignore_warnings : bool) : bool

The function opens a file in a format different from RTD. When opening the file, the program uses the current values of parameters read earlier from the file by means of the ReadExtFileParams function. The parameters are divided into two sets: FileInsertParams- parameters describing the manner of inserting the structure defined in the external file to the current project, and ExtFileParams - set of variables defined in the external file to provide a parametrized structure description. The function returns a non-zero value (True) if the file is opened correctly and inserted into the project. It returns a zero value (False) in the case of errors during file opening.

ReadExtFileParams (file_path : string, file_format : IRobotExternalFileFormat) : bool

The function reads variables from the indicated external file in a non-RTD format. The values of the variables will be used during the process of opening the file by means of the OpenExtFile function. Once the variables are read by means of the function, two sets of parameters are available: FileInsertParams - parameters describing the manner of inserting the structure defined in the external file to the current project, and ExtFileParams - set of variables defined in the external file to provide a parametrized structure description. The value of each parameter from each set may be modified and the modification will be recognized during opening the file by means of the OpenExtFile function. The function returns zero (False) in the case of errors during analysis of the indicated external file.

SaveAsExtFile (file_path : string, format : IRobotExternalFileFormat)

The function saves the current project to the file in the indicated external format (other than RTD).

NewFromTemplate (_tmpl_file : string)

Function creates a new document on the basis of a template defined by a specified file.

Available since version 2.0.

InsertExtFile (_file_path : string, _file_format : IRobotExternalFileFormat, _ignore_warnings : bool, _only_geometry : bool) : bool

The function inserts a structure defined in the external format (non-RTD format) file to the current project. Values of parameters describing definition and method of structure insertion may be read out (by means of the ReadExtFileParams function) and changed earlier.
If the structure is inserted correctly to the current project, then the function returns value different from zero (True).
Available since version 2.5.

SaveToFormat (_file_format : IRobotProjectSaveFormat, _file_path : string) : bool
Function saves the project to a file of the selected format.
Available since version 3.

IRobotProjectEvents

Events generated by the RobotProject object.
Available since version 7.5.

Operations:

OnSave (_file_path : string, _is_modified : bool)
Event generated directly before saving the project to the file of the specified path.
Available since version 7.5.

OnClose (_is_modified : bool)
Event generated directly before closing the document.
Available since version 7.5.

IRobotFileInsertParams

The proces of inserting a structure, defined in an external file of a non-RTD format, is carried out according to parameters determining the insertion mode. The RobotInsertParams interface provides access to the parameters determining the manner of inserting the structure into the current project.

Properties:

ReferenceNode : long
node number - structure insertion point

ScaleFactor : double
scaling coefficient

AsObject : bool
the flag indicating if the structure being inserted is to be treated as an object

Operations:

SetInsertionPoint (x : double, y : double, z : double)
The function sets the coordinates of the insertion point.

GetInsertionPoint (x : double*, y : double*, z : double*)
The function sets the coordinates of the insertion point.

SetRotation (alpha : double, beta : double, gamma : double)
The function allows one to determine the angle of the iserted structure.

GetRotation (alpha : double*, beta : double*, gamma : double*)
The function provides access to the rotation angle defined for the inserted structure.

IRobotProjectSaveFormat

Available formats for project saving.
Available since version 3.

Properties:

I_PSF_RTD : = 0

Available since version 3.

I_PSF_RTD_NORESULTS : = 1

Available since version 3.

I_PSF_STR : = 2

Available since version 3.

I_PSF_DXF : = 3

Available since version 3.

I_PSF_DXF_V14 : = 4

Available since version 3.

I_PSF_DWG : = 5

Available since version 3.

I_PSF_ANF : = 6

Available since version 3.

I_PSF_WRL : = 7

Available since version 3.

I_PSF_S : = 8

Available since version 3.

I_PSF_SAT : = 9

Available since version 3.

I_PSF_STP_CIM : = 10

Available since version 3.

I_PSF_STP_DSTV : = 11

Available since version 3.

Job preferences

IRobotProjectPreferences

Interface describing the settings for the current project (job)

Properties:

SectionsFound : IRobotSectionDatabaseList
the list of available (found) names for the section database

SectionsActive : IRobotSectionDatabaseList
the list of active (selected) section databases

Materials : IRobotMaterialDatabase
active material database

MeshParams : IRobotMeshParams
Meshing parameters
Available since version 3.

VehiclesFound : IRobotVehicleDatabaseList
List of names of available (found) vehicle databases
Available since version 3.

VehiclesActive : IRobotVehicleDatabaseList
List of names of active (selected) vehicle databases
Available since version 3.

Units : IRobotUnitMngr
object managing units and formats
Available since version 3.

MeshAutoAdjust : bool
automatic adjustment of finite element mesh
Available since version 4.5.

Operations:

GetActiveCode (code_type : IRobotCodeType) : string
The function returns the name of the active code of the indicated type.

SetActiveCode (code_type : IRobotCodeType, _code_name : string) : bool
The function sets the active code of the indicated type. If selection of an indicated code is not possible, then, zero value is returned (False).

Save ()
Function forces saving changes in the project settings.
Available since version 4.5.

GetActiveCodeNumber (_code_type : IRobotCodeType) : long
Function returns the numerical identifier of an active code of a given type.
Available since version 6.5.

SetActiveCodeNumber (_code_type : IRobotCodeType, _code_number : long) : bool
Function sets an active code of a given type based on the specified numerical identifier of the code.
Available since version 6.5.

IRobotSectionDatabaseList

The interface describing a list of section databases

Properties:

Count : int
number of section databases in the list

Operations:

Get (idx : int) : string
The function returns the name of section database saved in the list under the indicated index. The list elements are indexed from 1 to Count.

Find (db_name : string) : int

The function returns the list index of the database with the indicated name. If the required database is absent from the list, the function returns zero.

ChangeIndex (cur_idx : int, new_idx : int)
The function imposes a new position in the list for the selected database. By means of the function one may define a required order of section databases in the list. The order is important while searching through the databases for a section with a given name.

Add (db_name : string) : bool
The function adds a section database with a given name at the end of the list. It returns the message about carrying out the operation successfully

AddFromFile (file_path : string) : bool
The function adds a section database defined in the indicated file (.mdb) at the end of the list. It returns the message about carrying out the operation successfully

Remove (idx : int)
The function deletees a database from the list.

GetDatabase (idx : int) : IRobotSectionDatabase
The function returns the section database whose name is saved in the list in the indicated position.

IRobotMaterialDatabase

The interface providing access to material database

Properties:

Name : string
material database name

Operations:

GetAll () : IRobotNamesArray
The function returns a table of names of all materials defined in the database

Get (mater_type : IRobotMaterialType) : IRobotNamesArray
The function returns a table of names of materials of the indicated type that are defined in the database

LoadFromFile (file_path : string) : bool
The function reads the materials from the indicated file. If there occurs an error during the operation of reading from the database, the function returns zero (False).

Load (dbase_name : string) : bool
The function reads materials from the database with the indicated name. If there occurs an error during the operation of reading from the database or the database with the name does not exist, the function returns zero (False).

GetDefault (_mater_type : IRobotMaterialType) : string
Function returns the name of a selected type of default material.
Available since version 3.

SetDefault (_mater_type : IRobotMaterialType, _mater_name : string)
Function sets the name of a selected type of default material.
Available since version 3.

IRobotCodeType

Set of identifiers determining the types of codes that may be set in the Robot program.

Properties:

I_CT_STEEL_STRUCTURES : = 0
the code for steel structures

I_CT_STEEL_CONNECTIONS : = 1
the code for steel connections

I_CT_TIMBER_STRUCTURES : = 2
the code for timber structures

I_CT_RC_REAL_REINF : = 3
the code for real reinforcement

I_CT_RC_THEORETICAL_REINF : = 4
the code for theoretical reinforcement

I_CT_FOUNDATIONS_DESIGN : = 5
code for spread footing design

I_CT_CODE_COMBINATIONS : = 9
code combinations

I_CT_SNOW_WIND_LOADS : = 6
code for snow/wind loads

I_CT_SEISMIC_LOADS : = 7
code for seismic loads

I_CT_MOVING_LOADS : = 8
code for moving loads

IRobotSectionDatabase

The interface providing access to section database.

Properties:

Name : string
section database name

FullName : string
full name of section database
Available since version 4.

Description : string
Section database description
Available since version 7.5.

Operations:

GetAll () : IRobotNamesArray
The function returns a table containing names of all sections described in the database.

Load (db_name : string) : bool
The function reads sections from the section database with the indicated name. If such section database does not exist or there occurs an error during the reading process, the function returns zero.

LoadFromFile (file_path : string) : bool
The function reads sections from the database saved in the indicated file. If there occurred an error during the reading process, the function returns zero.

IRobotProjectPreferencesEvents

Events related to the change of settings for the current project.

Operations:

OnDialogOK ()
Event generated after accepting changes made in the project settings dialog box by the user.

Units and formats

Available since version 3.5.

IRobotUnitType

Unit types.
Available since version 3.5.

Properties:

I_UT_STRUCTURE_DIMENSION : = 1

Available since version 3.5.

I_UT_SECTION_DIMENSION : = 2

Available since version 3.5.

I_UT_SECTION_PROPERTIES : = 3

Available since version 3.5.

I_UT_STEEL_CONECTIONS : = 4

Available since version 3.5.

I_UT_DIAMETER_RC_BASE : = 5

Available since version 3.5.

I_UT_REINFORCEMENT_AREAS : = 6

Available since version 3.5.

I_UT_FORCE : = 7

Available since version 3.5.

I_UT_MOMENT : = 8

Available since version 3.5.

I_UT_STRESS : = 9

Available since version 3.5.

I_UT_DISPLACEMENT : = 10

Available since version 3.5.

I_UT_ANGLE_ROTATION_DATA : = 11

Available since version 3.5.

I_UT_ANGLE_ROTATION_RESULT : = 12

Available since version 3.5.

I_UT_TEMPERATURE : = 13

Available since version 3.5.

I_UT_WEIGHT : = 14

Available since version 3.5.

I_UT_MASS : = 15

Available since version 3.5.

I_UT_DIMENSIONLESS_QUALITY : = 16

Available since version 3.5.

I_UT_RULER : = 17

Available since version 3.5.

IRobotUnitData

Unit definition interface.
Available since version 3.5.

Properties:

Precision : long
precision of unit presentation - number of decimal places
Available since version 3.5.

E : bool
type of unit presentation: true value - exponential presentation, false value - decimal presentation
Available since version 3.5.

Name : string
unit name
Available since version 3.5.

Type : IRobotUnitType
unit type
Available since version 3.5.

IRobotUnitEditionType

Types of editable units.
Available since version 3.5.

Properties:

I_UMT_LENGTH : = 0

Available since version 3.5.

I_UMT_FORCE : = 1

Available since version 3.5.

I_UMT_MASS : = 2

Available since version 3.5.

IRobotUnitMngr

Interface of unit support management.
Available since version 3.5.

Properties:

UnitEdition : IRobotUnitEditionServer
edition server
Available since version 3.5.

Operations:

Count (_type : IRobotUnitType) : long
Function returns the number of available units of the specified type.
Available since version 3.5.

Set (_type : IRobotUnitType, _data : IRobotUnitData)
Function determines active unit of the specified type based on the provided definition.
Available since version 3.5.

Get (_type : IRobotUnitType) : IRobotUnitData
Function takes definition of the active unit for a given type.
Available since version 3.5.

GetName (_type : IRobotUnitType, _idx : long)
Function takes name of the unit of the specified type with the determined index.
Available since version 3.5.

GetCoeff (_unit_type : IRobotUnitType) : double
The function returns the coefficient of conversion of the currently-selected unit of a given type to the base unit.

Refresh ()
Function forces refreshing current information about units in the whole application.

IRobotUnitComplexData

Interface of composed unit definition. A composed unit is a unit being a function of other two units. For example, unit of the moment is a product of force unit and length unit.
Available since version 3.5.

Properties:

Name2 : string
Name of the second component unit.
Available since version 3.5.

IRobotUnitEditionServer

Interface enabling unit edition.
Available since version 3.5.

Operations:

Get (_type : IRobotUnitEditionType, _idx : long) : IRobotUnitEditionData
Function takes the edit definition of unit of the specified type with the specified index.
Available since version 3.5.

Set (_data : IRobotUnitEditionData)
Function updates unit definition.
Available since version 3.5.

Find (_type : IRobotUnitEditionType, _name : string) : long
Function returns index of the unit of the specified type with the determined name.
Available since version 3.5.

New (_type : IRobotUnitEditionType, _unit : name, _coeff : double) : long
Function generates a new edit unit based on given parameters.
Available since version 3.5.

Delete (_type : IRobotUnitEditionType, _idx : long)
Function deletes a unit of the specified type with the determined index.
Available since version 3.5.

Count (_type : IRobotUnitEditionType) : long
Function returns the number of units of the specified type.
Available since version 3.5.

IRobotUnitEditionData

Interface of unit edit definition. An edit definition is a definition which determines a base unit and a proportionality factor.
Available since version 3.5.

Properties:

Unit : string
name of base unit
Available since version 3.5.

Coefficient : double
number determining the ratio of defined unit to base unit
Available since version 3.5.

Type : IRobotUnitEditionType
type of edit unit
Available since version 3.5.

IRobotUnitMngrEvents

Events (notifications) generated by the unit manager.

Operations:

UnitsChanged ()
Event generated after any change in the settings of units.

Project components

A project may contain many components of different types. They are stored in a tree-like hierarchical structure and managed by an appropriate component manager.

IRobotProjectComponentType

A set of identifiers is defined to describe different types of components that may be saved in a project.

Properties:

I_PCT_BEAM : int = 3
concrete beam

I_PCT_COLUMN : int = 4
concrete column

I_PCT_FOOT : int = 5
concrete footing

I_PCT_JOINT : int = 6
steel connection

I_PCT_DRAWING : int = 7
plotter drawing

I_PCT_BWALL : int = 8
concrete wall

I_PCT_RETAINING_WALL : = 10
Retaining wall
Available since version 3.

I_PCT_SLAB : = 9

Available since version 3.

I_PCT_CONTINUOUS_FOOT : = 11

Available since version 3.

IRobotProjectComponent

Common interface granting access to any project component.

Properties:

Type : IRobotProjectComponentType
project component type

Name : string
component name

Data : IDispatch
object providing access to data on a project component; the type of the object depends on the component type

IRobotProjectComponentMngr

Manager of project components manages tree-like structures containing all components defined in a project. Project components are divided into different levels. The manager makes available a list of all levels that are indexed with numbers from 1 to the number of levels. Additionally, each level has its name. Project components may be taken by levels. It is possible to take from the project all the components of an indicated type from the level of the indicated name.

Properties:

LevelCount : long
number of levels in the structure containing all project components.

StdLevelName : string
name of standard level
Available since version 2.0.

Operations:

GetLevelName (level_idx : long) : string
The function returns the name of the level with the indicated index. Levels are indexed with numbers from 1 to LevelCount.

Get (cmpnt_type : IRobotProjectComponentType, level_name : string) : IRobotCollection
The function returns a collection of components of the indicated type from the level of the indicated name. Collection components are objects of the RobotProjectComponent type.

Create (cmpnt_type : IRobotProjectComponentType, cmpnt_name : string, level_name : string) : IRobotProjectComponent
Function creates a project component of the defined type and locate it on a specified level.
Available since version 2.0.

Concrete footing

A set of interfaces allowing access to a component of the 'spread footing' project.

Available since version 3.

IRConcrFooting

Interface representing the spread footing.

Available since version 3.

Properties:

Geometry : RConcrFootingGeometry
Footing geometry
Available since version 3.

Ground : RConcrFootingGround
soils under the spread footing
Available since version 3.

Loads : RConcrFootingLoads
spread footing loads and backfill loads
Available since version 3.

Results : RConcrFootingResults
Analysis results for spread footing
Available since version 3.

Operations:

Calculate ()
Function runs complete spread footing calculations.
Available since version 3.

IRConcrFootingGeometry

Interface representing the spread footing geometry.
Available since version 3.

Properties:

Shape : RConcrFootingShapeType
shape of the spread footing section
Available since version 3.

Type : RConcrFootingType
Foundation type
Available since version 3.

Operations:

GetDim (_dimType : RConcrFootingDimType) : double
Function returns the value of a relevant spread footing dimension.
Available since version 3.

SetDim (_dimType : RConcrFootingDimType, _val : double)
Function sets the value of a relevant spread footing dimension.
Available since version 3.

IRConcrFootingDimType

A set of values identifying individual spread footing dimensions.
Available since version 3.

Properties:

I_CFDT_FOOT_DIM_A : = 1

Available since version 3.

I_CFDT_FOOT_DIM_B : = 2

Available since version 3.

I_CFDT_FOOT_DIM_H1 : = 3

Available since version 3.

I_CFDT_FOOT_DIM_H2 : = 4

Available since version 3.

I_CFDT_FOOT_DIM_H3 : = 5

Available since version 3.

I_CFDT_FOOT_DIM_H4 : = 6

Available since version 3.

I_CFDT_FOOT_DIM_EX : = 7

Available since version 3.

I_CFDT_FOOT_DIM_EY : = 8

Available since version 3.

I_CFDT_FOOT_DIM_AP : = 9

Available since version 3.

I_CFDT_FOOT_DIM_BP : = 10

Available since version 3.

I_CFDT_FOOT_DIM_L : = 11

Available since version 3.

I_CFDT_FOOT_DIM_COL_A : = 12

Available since version 3.

I_CFDT_FOOT_DIM_COL_B : = 13

Available since version 3.

IRConcrFootingShapeType

A set of values defining the spread footing shape.
Available since version 3.

Properties:

I_CFST_FOOT_SHAPE_RECT : = 1
spread footing of a rectangular section
Available since version 3.

I_CFST_FOOT_SHAPE_TRAPEZOIDAL : = 2
spread footing of a trapezoidal section
Available since version 3.

IRConcrFootingType

A set of values defining the foundation type.
Available since version 3.

Properties:

I_CFT_FOOT_TYPE_FOOTING : = 1
column footing
Available since version 3.

I_CFT_FOOT_TYPE_WALL : = 2
continuous footing
Available since version 3.

IRConcrFootingGround

Interface representing soils and reference level of the footing.
Available since version 3.

Properties:

SoilLevel : double

reference level of the footing
Available since version 3.

ColumnPierLevel : double
Column pier level
Available since version 3.

IRConcrFootingLoads

Interface representing spread footing loads and backfill loads.
Available since version 3.

IRConcrFootingResults

Interface representing calculation results for the spread footing.
Available since version 3.

Properties:

Ax : double
Required reinforcement area (X direction)
Available since version 3.

Ay : double
Required reinforcement area (Y direction)
Available since version 3.

Support of external file formats

A structure and an entire project is saved to the file (RTD format). It is, however, possible to open and save structures defined in other external file formats.

IRobotExternalFileFormat

A set of identifiers is defined to describe different external file formats accepted by Robot.

Properties:

I_EFF_STR : = 1
text file STR format

I_EFF_DXF : = 2
DXF format

IRobotSTRParams

Interface granting access to the set of parameters (variables) defined by the STR format files.

Properties:

Count : long
number of managed parameters

Operations:

GetParameter (idx : long) : IRobotSTRParameter
The function returns a structure describing a parameter with the indicated index

FindParameter (param_name : string) : long
The function returns the index associated with the parameter of the indicated name

IRobotSTRParameter

The interface describing the variable of text file in STR format.

Properties:

Type : IRobotSTRParamType
variable type

Name : string
variable name

Description : string
variable description

IsActive : bool
flag indicating if the variable is active

IntegerVal : long
value of the variable of the I_STR_PT_INTEGER type

DoubleVal_1 : double
value of the variable of the I_STR_PT_DOUBLE type. It is also the first out of three values of the variable of the I_STR_PT_DOUBLE_3 type.

DoubleVal_2 : double
the second out of three variables of the I_STR_PT_DOUBLE_3 type

DoubleVal_3 : double
the third out of three variables of the I_STR_PT_DOUBLE_3 type

TextVal : string
value of the variable of the I_STR_PT_TEXT type. It is also the value currently selected from the list for the variable of the I_STR_PT_TEXT_LIST type.

TextList : string
string of characters containing all possible values (separated by commas) of the variable of the I_STR_PT_TEXT_LIST type.

SelectionVal : string
value of the variable of the I_STR_PT_SELECTION type describing a selection by means of text

FilePathVal : string
value of the variable of the I_STR_PT_GROUP type representing the access path to the file with the appropriate bitmap.

IRobotSTRParamType

The Robot text file allows one to define variables of different types. Depending on variable type, they may assume values of different character. A set of identifiers is defined to refer to different types of variables available in STR files.

Properties:

I_STR_PT_INTEGER : = 1
variable assuming the value in the form of an integer

I_STR_PT_DOUBLE : = 2
variable assuming the value in the form of a real number

I_STR_PT_TEXT : = 3
variable assuming the value in the form of a text

I_STR_PT_SELECTION : = 4
variable assuming the value in the form of a text describing selection

I_STR_PT_DOUBLE_3 : = 5
variable assuming the value described by means of three real numbers

I_STR_PT_TEXT_LIST : = 6
variable assuming the value in the form of a specially formated text. It consists of fragments separated by commas, defining the list of possible text values, and the (repeated) currently-selected fragment.

I_STR_PT_SECTION : = 7
variable containing the section name (as a text)

I_STR_PT_GROUP : = 8
variable representing a group of varibles. Its value consists of two elements: group name (stored in the TextVal variable), and the path to the file containing the bitmap associated with the group (FilePathVal)

Data Server

All types of data used to define and model a structure (nodes, bars, load cases)) are managed by the Robot Object Model and the Data Server. The server also provides access to the results of calculations carried out for the defined structures.

IRobotStructure

RobotStructure represents an entire structure. Particular structure components are represented by appropriate interface components.

Properties:

Labels : IRobotLabelServer
server of complex attributes, such as node supports, bar sections and others.

Nodes : IRobotNodeServer
Structure node server

Bars : IRobotBarServer
structure bar server

Cases : IRobotCaseServer
load case server

Results : IRobotResultServer
server of the currently available results

Selections : IRobotSelectionFactory
server managing selections (selection factory)

Objects : IRobotObjObjectServer
object server

FiniteElems : IRobotFiniteElementServer
Finite element server
Available since version 2.5.

Groups : IRobotGroupServer
server managing groups of structure components
Available since version 3.

Edit : IRobotStructureEditTools
set of auxiliary edit functions
Available since version 3.5.

ResultsFreeze : bool

Available since version 4.

Storeys : IRobotStoreyMngr
stories defined in a structure

GroupObjects : IRobotGroupObjectServer
server for grouping of structure elements
Available since version 7.5.

QuantitySurvey : IRobotStructureQuantitySurvey
quantity survey
Available since version 7.5.

Type : IRobotProjectType
structure type
Available since version 8.

Operations:

CreateCache () : IRobotStructureCache

Function creates and returns an empty buffer to which new structure components can be added. To insert a whole component set to a structure, ApplyCache function should be called up.
Available since version 3.5.

ApplyCache (_struct_cache : IRobotStructureCache) : IRobotStructureApplyInfo
Function inserts data from a specified buffer to a structure and returns information about new structure components generated in this way.
Available since version 3.5.

ExportXml (_input_xml : string, _output_xml : string)
Function exports selected structure components and calculation results to the indicated xml file.
Available since version 4.1.

Clear ()
Function deletes all elements of a structure model.

Merge (_data : IRobotStructureMergeData, _params : IUnknown)
Function attaches a given data set to the curent structure.
Available since version 8.

IRobotStructureApplyInfo

Interface providing access to information about structure components generated as a result of inserting a new data set to a structure.
Available since version 3.5.

Properties:

Nodes : IRobotNumbersArray
table with node numbers
Available since version 3.5.

Bars : IRobotNumbersArray
table with bar numbers
Available since version 3.5.

IRobotStructureCache

Interface enabling buffered access to a structure. It enables adding quickly a whole component set to a structure.
To add a whole data set to a structure, ApplyCache function of RobotStructure inerface should be called up.
Available since version 3.5.

Operations:

AddNode (_number : long, _x : double, _y : double, _z : double)

Function adds a new node to the buffer.
Available since version 3.5.

AddBar (_number : long, _start_node : long, _end_node : long, _sect_name : string, _mater_name : string, _gamma : double = 0)
Function adds a new bar to the buffer.
Available since version 3.5.

SetBarLabel (_bar : long, _lab_type : IRobotLabelType, _lab_name : string, _ext_par : long = 0)
Function enables assigning a specified label to an earlier-generated bar.
Available since version 4.1.

EnsureNodeExist (_x : double, _y : double, _z : double) : long
Function checks if a node exists at the position of defined coordinates. If yes, then function returns its number. If no, then function creates a new node and returns its number.

IRobotStructureMergeData

Interface defining data that may be merged into a structure by means of the Merge function.
Available since version 8.

Properties:

Structure : IRobotStructure
auxiliary structure
Available since version 8.

Operations:

CreateStructure (_type : IRobotProjectType) : bool
Function creates an empty auxiliary structure of the specified type.
Available since version 8.

LoadStructure (_rtd_fpath : string) : bool
Function loads an auxiliary structure from the specified RTD file.
Available since version 8.

Basic definitions

Data Server manages a large amount of different types of data describing structures. In order to simplify and unify the manipulation of the data, certain basic abstractions have been defined. They allow one to classify the data. Each structure component is represented in the Data Server by means of an object of the appropriate type. All objects of the same type have the same set of attributes and the same functionality described with the set of operations that may be performed on the object. The set of

attributes and operations is described formally by means of the appropriate interface. For instance, attributes and complete functionality of a structure node is described by the RobotNode interface. The object type determines univocally how it is possible to work with the object in question, and it defines the access interface for object attributes and functionalities.

Uniform access to structure components

In order to unify access mode to different structure components, the following two notions have been defined: Object (RobotDataObject) and Object Server (RobotDataObjectServer). An Object represents any structure component. The Object Server manages all objects of the same type. For instance, a structure node is represented in the Robot Object Model s an object of the RobotNode type, and all nodes are managed (created, made accessible, deleted) by the RobotNodeServer type.

IRobotDataObject

RobotDataObject defines a common interface for structure components. Each structure component has its own user-defined number that is unique among all the components of the same type. The user may use the number to identify objects. There is one uniform manner of ascribing and accessing complex attributes (labels) defined for all structure components.

Properties:

Number : long

object number is assigned by the user; it is qunique among all the components of the same type.

Operations:

SetLabel (label_type : IRobotLabelType, label_name : string)

The function applies the defined label to an object. At a given moment, an object may have only one label of a type applied (e.g. there may not be two supports defined for one node). If an object has a formerly applied label of the same type, it is replaced by the new one.

HasLabel (label_type : IRobotLabelType) : bool

The function returns True (non-zero value) if a label of the type has already been defined for the object.

RemoveLabel (label_type : IRobotLabelType)

The function deletes a label of the indicated type.

GetLabelName (label_type : IRobotLabelType) : string

The function returns the name of a label of the indicated type, applied to the object. If the object does not have any label of the type, running the function leads to a critical error.

GetLabel (label_type : IRobotLabelType) : IRobotLabel

The function returns a label of the indicated type applied to the object. If the object does not have any label of the type, running the function leads to a critical error.

GetLabels () : IRobotCollection

The function returns a collection containing all labels (of different types) defined for a given object.

IRobotDataObjectServer

All objects of the same type are managed by the same object server. RobotDataObjectServer defines a common interface for such servers.

Operations:

Exist (number : long) : bool

The function returns True (non-zero value) if the object of the indicated name already exists.

Get (number : long) : IRobotDataObject

The function returns an object with the indicated user-defined number. The type of the returned object agrees with the type of objects managed by the server. If the object of the indicated number does not exist, running the function leads to critical error.

GetMany (selection : IRobotSelection) : IRobotCollection

The function returns a collection of objects that meet the criteria of the indicated selection.

GetAll () : IRobotCollection

The function returns a collection containing all objects managed by the server.

Delete (number : long)

The function deletes the object of the indicated number.

DeleteMany (selection : IRobotSelection)

The function deletes all objects that meet the criteria of the indicated selection.

SetLabel (object_selection : IRobotSelection, label_type : IRobotLabelType, label_name : string)
The function applies a label (identified by the type and name) to all objects that meet the criteria of the indicated selection.

RemoveLabel (object_selection : IRobotSelection, label_type : IRobotLabelType)
The function removes the labels of the indicated type from all objects that meet the criteria of the indicated selection.

IRobotObjectType

RobotObjectType defines identifiers for all types of objects managed by the Data Server

Properties:

I_OT_NODE : int = 0
node

I_OT_BAR : int = 1
bar

I_OT_PANEL : int = 4
panel

I_OT_FINITE_ELEMENT : int = 5
finite element

I_OT_GEOMETRY : int = 6
geometrical object

I_OT_FAMILY : int = 3
group

I_OT_CASE : int = 2
load case

I_OT_OBJECT : = -2
shared auxiliary identifier for the following types of structure components: bar, slab, volumetric object and geometric object

Available since version 2.0.

I_OT_UNDEFINED : = -1
auxiliary identifier denoting undefined or unknown object type
Available since version 2.0.

I_OT_VOLUME : = 7
Volumetric object
Available since version 3.

Querying mechanism

Data Server defines a simple querying mechanism that allows one to select only the relevant structure elements from among all the structure elements managed by the server. The selection process consists of two phases: determining the structure elements that are of interest for the designer, and taking them from the server and performing an operation on the elements. Selection defined by means of the object of RobotSelection type is used to determine the relevant sub-set of objects - structure components. Thus, a new selection is to be defined by creating a new object of RobotSelection type. By means of the operations available for an object of RobotSelection type, one may describe the selection of the relevant structure components. On the basis of the RobotSelection object, one may obtain a set of objects that meet the criteria determined in Selection. The mode of access to such a set is defined by means of the Collection described formally by the RobotCollection type. The collection simplifies the access to all its elements, orders and indexes them with numbers from 1 to the number of elements in the collection. Thus, it is easy to browse through the objects belonging to the collection. A collection created on the basis of the formerly defined selection allows one to get access to all the object that meet the selection criteria.

IRobotSelection

RobotSelection allows one to define selections of particular structure components

Properties:

Type : IRobotObjectType
types of objects described by selections (e.g. I_OT_NODE for node selection). The object type is applied when a selection is created and it may not be changed.

Operations:

FromText (sel_text : string)
Robot allows one to define a selection of particular structure components by introducing a text - list describing the selection in the appropriate syntax. Such text may also be used as a parameter of the FromText method that creates on the basis of

the text the relevant selection. A text provided as the parameter should describe the selection in the currently used interface language of Robot.

ToText () : string
Returns a selection description in the form of the appropriate text

Clear ()
Resetting selection - selection is empty after this operation

Add (sel : IRobotSelection)
Adding the indicated selection to the current one.

Exclude (sel : IRobotSelection)
Logical complement of the current selection and the selection indicated as the argument

And (sel : IRobotSelection)
Operation of logical product with the selection provided as the argument

IRobotCollection

The RobotCollection type is defined to unify the access mode to the sub-set of objects of a given type. Each object belonging to a collection is linked with a number - the index of the object in the collection. In order to obtain an object, one should provide its index. Indexes assume values from 1 to Count, where Count is the number of all objects in the collection. One should differentiate between the object index and the user-defined object number (e.g. node number). The user-defined number of an object is ascribed permanently to the object, unless the user changes the number. The index, in turn, corresponds to the position of the object in the given collection - the same object may have different indexes in different collections. Indexes are ascribed to objects in a given collection in such a way that the increasing index values correspond to the increasing values of user-defined numbers ascribed to the objects. For instance, if a collection consists of three objects with the user-defined numbers 24, 105 and 214, the objects obtain the following indexes in the same collection: 1, 2, 3.

Properties:

Count : long
Number of all objects in a collection (it is also the index of the last element of the collection)

Operations:

Get (index : long) : IDispatch

The function returns an object with the indicated index in the collection. The type of returned object depends on the collection type - for a collection of nodes, the Get method returns an object of the RobotNode type.

IRobotMultiSelection

RobotMultiSelection interface describes a set of many selections of different types. Multi-selection may be useful to describe complex selections of objects of more than one type.

Properties:

Modes : IRobotModeSelection

mode selection is treated as a supplement to load case selection
Available since version 3.5.

CaseCmpnt : long

selected number of a load case component (it refers only to code combinations);
zero value indicates selection of all components (selection is not limited to a specific component)
Available since version 3.5.

Operations:

Exist (obj_type : IRobotObjectType) : bool

The function returns True (non-zero value) if the multi-selection contains the selection of objects of the indicated type.

Set (obj_type : IRobotObjectType, sel : IRobotSelection)

The function sets the indicated selection as the selection of objects of a given type within the multi-selection

Get (obj_type : IRobotObjectType) : IRobotSelection

The function returns selection of objects of the indicated type

IRobotMultiCollection

The RobotMultiCollection groups many collections of objects of various types.

Operations:

Exist (obj_type : IRobotObjectType) : bool

The function returns True (non-zero value) if there is a collection of objects of the indicated type within the multi-collection

Set (obj_type : IRobotObjectType, collection : IRobotCollection)
The function sets the indicated collection as the collection of objects of the indicated type within the multi-collection

Get (obj_type : IRobotObjectType) : IRobotCollection
The function returns a collection of objects of a similar type, contained in the given multi-collection.
Before selecting a collection, one should check by means of the Exist function if it has been defined within the framework of the multi-collection.

IRobotModeSelection

Interface describing selection of a mode for a modal load case.
Available since version 3.5.

Properties:

Type : IRobotModeSelectionType
type of mode selection
Available since version 3.5.

Mode : long
mode number - its interpretation depends on a chosen type of mode selection
Available since version 3.5.

Combination : IRobotModeCombinationType
selected type of a combination which includes all load case modes
Available since version 3.5.

Labels - complex attributes of objects

The notion of Label has been introduced to simplify the description of complex object attributes - structure components.
A complex attribute is an attribute for the description of which a set of values is required, as opposed to a simple attribute described by one value. A support is an example of a complex attribute. In order to define a support, one should provide an entire set of values describing particular support parameters. On the other hand, the entire support is an attribute of a supported node. The set of all values describing particular quantities-values of a complex attribute becomes in a sense the value of this attribute. A label is used to name and identify such a complex value. It suffices to provide once the values of all attribute parameters (e.g. a support) and associate a name with this set (a character string), to make it possible to use the name only instead of the entire set of values. Thus, in order to apply an appropriate support to a node, it suffices to provide the name of a support (the label associated with the set of values describing all support parameters) and it is not necessary to reinforcement-introduce the values of particular support parameters.

Since there are many different types of complex attributes, the notion of a complex attribute i.e. a label type (RobotLabelType) has been introduced. The name of a label is unique within the set of labels of the same type.

IRobotLabelType

A set of Label types available in the Data Server is defined.

Properties:

I_LT_NODE_SUPPORT : int = 0
label describing a node support

I_LT_NODE_RELEASE : int = 1
label describing a node release

I_LT_NODE_COMPATIBILITY : int = 2
label describing a compatibility

I_LT_BAR_SECTION : int = 3
label describing a bar section

I_LT_BAR_RELEASE : int = 4
label describing a bar release

I_LT_BAR_OFFSET : int = 5
label describing a bar offset

I_LT_BAR_CABLE : int = 6
label describing a cable

I_LT_BAREND_BRACKET : int = 7
label describing a bracket at one of bar ends

I_LT_BAR_MATERIAL : int = 8
label describing a material

I_LT_EDGE_SUPPORT : = 10
label describing support defined at the edge

I_LT_PANEL_THICKNESS : = 11
label describing a plate thickness

I_LT_PANEL_REINFORCEMENT : = 12
label describing a plate reinforcement

I_LT_UNKNOWN : = -1
unknown label type
Available since version 1.7.

I_LT_SUPPORT : = 0
Label describing a support
Available since version 2.0.

I_LT_MATERIAL : = 8
Label describing a material
Available since version 2.0.

I_LT_LINEAR_RELEASE : = 13
linear release between panels
Available since version 2.5.

I_LT_BAR_ELASTIC_GROUND : = 14
Elastic ground (bars)
Available since version 2.5.

I_LT_NODE_RIGID_LINK : = 15
rigid link
Available since version 2.5.

I_LT_MEMBER_TYPE : = 16
member type for steel/aluminum and timber design
Available since version 2.5.

I_LT_VEHICLE : = 17
Vehicle definition
Available since version 3.

I_LT_SOLID_PROPERTIES : = 18
Solid properties
Available since version 3.

I_LT_BAR_GEO_IMPERFECTIONS : = 19
Geometrical imperfections
Available since version 3.

I_LT_BAR_NONLINEAR_HINGE : = 20
Non-linear hinge
Available since version 3.

I_LT_CLADDING : = 21

Available since version 3.5.

IRobotLabel

The RobotLabel interface describes the complex attribute referred to as Label in the Robot Object Model. The complex attribute is defined by an entire set of values and not by one value, as in the case of a simple attribute. The set of values is associated with a name - a character string. The set of values describing complex attribute parameters consists the data of the label, while the character string associated with the set consists the label name.

Properties:

Type : IRobotLabelType
label type

Name : string
name associated with the label - the name us unique among all the labels of the same type and therefore it may be used to identify the label.

Data : IDispatch
object containing the label data - set of values describing the complex attribute represented by the label. The object type depends on the label type. For instance, the data of a label describing a node support (i.e. labels of the I_LT_NODE_SUPPORT type) are represented by the object of the RobotNodeSupportData type.

UniqueId : long
unique label identifier
Available since version 1.7.

IRobotLabelServer

All labels are managed (created, made accessible, deleted) by the label server. Irrespective of label type, a label must be defined by means of the server to be used in structure definition later on.

Operations:

Create (label_type : IRobotLabelType, label_name : string) : IRobotLabel
The function creates and returns a new label of the defined type and with the defined name. if labels of the indicated type are stored in a database (as in the case foundation bar section type labels - I_LT_BAR_SECTION), then, the function Create looks through the currently available databases to fine a label with the defined name. If the label is not found in the database, all parameter values describing a label will be initialized with appropriate values from the database. If the user defines a label that is absent from the database, then, he should define appropriate values of all the parameters describing the complex attribute.

Store (label : IRobotLabel)

Once a label is created and the values of its particular parameters are defined, one should save it in the structure so that it could be applied to appropriate structure component. If a label of the same type and name has already been saved in the structure, the values of all its parameters will be up-dated on the basis of the newly-defined values.

Delete (label_type : IRobotLabelType, label_name : string)

The function deletes a label of the indicatd type and name. If the label has been used as an attribute value of a structure component, then, the value of the attribute is set as undefined after deleting the label. For instance, if a bar had STEEL defined as its material, then, after deleting the STEEL label of the I_LT_BAR_MATERIAL type from the label server, the bar will have no material defined.

Exist (label_type : IRobotLabelType, label_name : string) : bool

The function checks if there exist the label of the indicated type and name, i.e. if it has been saved in the structure.

Get (label_type : IRobotLabelType, label_name : string) : IRobotLabel

The function returns the label of the indicated type and name. If such label already exists, running the function leads to critical error.

GetMany (label_type : IRobotLabelType) : IRobotCollection

The function returns a collection containing all label of the indicated type saved in the structure

GetAll () : IRobotCollection

The function returns all labels of all types that have been saved in the structure.

FindWithId (unique_id : long) : IRobotLabel

Function finds and returns a label with the specified identifier. If such a label is not found, then an "empty reference" is returned (Nothing in the languages such as Visual Basic, NULL in the languages such as C++).
Available since version 1.7.

StoreWithName (_label : IRobotLabel, _label_name : string)

Function saves the indicated label under the indicated name.
Available since version 3.

GetAvailableNames (_label_type : IRobotLabelType) : IRobotNamesArray

Function returns the table with names of available labels of a given type. The available labels are those that may be assigned to structure elements by means of the SetLabel function. They include all the labels used earlier in the current structure and global labels - available in each structure.
Available since version 3.

IsAvailable (_lab_type : IRobotLabelType, _lab_name : string) : bool
Function returns a non-zero value (True) if the indicated label is available - it may be assigned to a structure element by means of the SetLabel function.
Available since version 3.

CreateLike (_lab_type : IRobotLabelType, _lab_name : string, _like_name : string) : IRobotLabel
Function creates and returns a new label of the indicated type with the indicated name. A data set defining the label will be copied from the label named the "_like_name" parameter.
Available since version 3.

SetDefault (_lab_type : IRobotLabelType, _lab_name : string) : bool
Function sets a name of a default label of the specified type.
Default labels are applied during automatic generation of structure elements e.g. as a result of edit operations performed on a structure.
Available since version 3.5.

GetDefault (_lab_type : IRobotLabelType) : string
Function returns the name of a default label of the specified type.
Available since version 3.5.

Selections

IRobotSelectionFactory

Factory that creates objects representing selections.

Operations:

Create (object_type : IRobotObjectType) : IRobotSelection
The function creates and returns an object representing selections of structure components of an indicated type. The newly-created selection is empty.

CreateMulti () : IRobotMultiSelection
The function creates and returns an empty object of the multi-selection type

Get (obj_type : IRobotObjectType) : IRobotSelection
The funcion gives the current selection of objects of a given type back.

CreateFull (obj_type : IRobotObjectType) : IRobotSelection
Function creates selection containing all objects of the specified type.
Available since version 1.7.

CreatePredefined (predefined_sel : IRobotPredefinedSelection) : IRobotSelection
Available since version 1.7.

CreateByLabel (_obj_type : IRobotObjectType, _label_type : IRobotLabelType, _label_name : string) : IRobotSelection
Function creates a selection of objects of the indicated type, which are assigned a label of the specified type and name.
Available since version 3.

IRobotModeSelectionType

A set of identifiers determining various manners of mode selection has been defined.

Properties:

I_MST_NONE : = 1
empty selection of modes

I_MST_ALL : = 2
selection containing all modes

I_MST_SINGLE : = 3
selection of one indicated mode

I_MST_FIRST_N : = 4
selection of the first n modes

IRobotModeCombinationType

A set of identifiers is defined to determine different types of combinations of modes.

Properties:

I_MCT_NONE : = -1
no combinations

I_MCT_SRSS : = 0
SRSS type combination

I_MCT_CQC : = 1
CQC type combination

I_MCT_10P : = 2
10 percent

I_MCT_2SM : = 3
double sum

I_MCT_ALL : = -2

Available since version 3.5.

IRobotPredefinedSelection

Properties:

I_PS_CASE_SIMPLE_CASES : = 2
selection of all simple cases
Available since version 1.7.

I_PS_CASE_COMBINATIONS : = 3
selection of all combination (but not code combinations)
Available since version 1.7.

I_PS_CASE_CODE_COMBINATIONS : = 4
selection of all code combination
Available since version 1.7.

I_PS_NODE_USER_NODES : = 21
User-defined nodes
Available since version 2.5.

I_PS_NODE_CALC_NODES : = 22
nodes defined automatically during generation of a calculation model
Available since version 2.5.

I_PS_NODE_SUPPORTED : = 23
nodes at which supports have been defined
Available since version 2.5.

I_PS_BAR_INACTIVE : = 31
inactive bars
Available since version 4.5.

IRobotGroup

A group of structure components of a given type.
Available since version 3.

Properties:

ObjectType : IRobotObjectType
type of objects belonging to the group
Available since version 3.

Name : string
Group name
Available since version 3.

SelList : string
list describing the objects belonging to the group
Available since version 3.

Color : long
Color assigned to a group
Available since version 3.

Operations:

CreateSelection () : IRobotSelection
Function creates and returns the selection describing the objects belonging to the group.
Available since version 3.

CreateCollection () : IRobotCollection
Function creates and returns the collection containing the objects belonging to the group.
Available since version 3.

Read ()
Function reads group definition from the structure model.
Available since version 3.

Store ()
Function saves the group definition in the structure model.
Available since version 3.

IRobotGroupServer

Object managing groups of structure elements.
Available since version 3.

Operations:

GetCount (_obj_type : IRobotObjectType) : long
Function returns the number of groups containing a given type of objects defined in the structure.
Available since version 3.

Get (_obj_type : IRobotObjectType, _idx : long) : IRobotGroup
Function returns the group containing a given type of objects of the specified index. Groups are indexed from 1 to the number of groups containing a given type of objects.
Available since version 3.

Find (_obj_type : IRobotObjectType, _grp_name : string) : long
Function returns index of the group of the given type of objects. If such a group does not exist, then zero value is returned.
Available since version 3.

Delete (_obj_type : IRobotObjectType, _idx : long)
Function deletes the group containing a given object type of the specified index.
Available since version 3.

Create (_obj_type : IRobotObjectType, _name : string, _sel_list : string = "", _color : long = 0) : long
Function creates a new group containing objects of the indicated type with the specified name. Function returns the index of the created group.
Available since version 3.

Structure components

All structure components are represented by the objects of relevant data types.

Nodes

A structure node is described by means of the RobotNode type. All the nodes are managed by the node server of the RobotNodeServer type.

IRobotNode

Derived from IRobotDataObject
Each structure node is represented in the Robot Object Model by means of the RobotNode interface.

Properties:

X : double
coordinate x

Y : double
coordinate y

Z : double
coordinate z

UniqueId : long
unique identifier of structure component

Available since version 1.7.

HasEmitter : bool
flag indicating if emitter is assigned to the node
Available since version 1.7.

IsCalc : bool
flag indicating if the node has been created automatically due to generation of the calculation model
Available since version 2.5.

HasCalcSupport : bool
Flag indicating that a support was created automatically at the node during generation of a structure model
Available since version 8.

Operations:

SetEmitter (emit : IRobotEmitter)
Function defines emitter in the node
Available since version 1.7.

GetEmitter () : IRobotEmitter
Function returns the emitter defined in the node. If the emitter has not been defined the function returns empty reference (NULL).
Available since version 1.7.

RemoveEmitter ()
Function deletes emitter from the node
Available since version 1.7.

GetCalcSupport () : IRobotNodeSupportData
Function makes available data defining a support created automatically at the node during generation of a structure model.
Available since version 8.

IRobotNodeServer

Derived from IRobotDataObjectServer
Server of nodes manages all the structure nodes

Properties:

FreeNumber : long
first available user number which can be assigned to the structure node
Available since version 1.7.

RigidLinks : IRobotNodeRigidLinkServer
rigid link server
Available since version 2.5.

CompatibleNodes : IRobotNodeCompatibilityServer
compatible node server
Available since version 2.5.

NonlinearLinks : IRobotNonlinearLinkServer
non-linear connection models
Available since version 3.

Operations:

Create (node_number : long, x : double, y : double, z : double)
The function creates a new node with the provided parameters.

FindWithId (unique_id : long) : long
Function returns a number of the node with a specified unique identifier. If such a node is not found, then, zero value is returned.
Available since version 1.7.

SetEmitter (node_sel : IRobotSelection, emit : IRobotEmitter)
Function defines a determined emitter on the specified node list.
Available since version 1.7.

RemoveEmitter (node_sel : IRobotSelection)
Function deletes emitter definition from nodes belonging to the specified selection.
Available since version 1.7.

IsCalc (_node_num : long) : bool
Function returns a non-zero value (True), if the node of the specified number has been created automatically during generation of the calculation model

Available since version 2.5.

GetUserNodes () : IRobotCollection
Function returns the collection of all user-defined nodes.
Available since version 2.5.

GetCalcNodes () : IRobotCollection
Function returns the collection of all nodes created automatically during generation of the calculation model.

Available since version 2.5.

GetConnectedBars (_node_sel : string) : IRobotNumbersArray
Function returns the table with numbers of bars adjoining the specified list of nodes.
Available since version 3.5.

HasCalcSupport (_node_number : long) : bool
Function returns True value if a support was automatically created at the specified node during generation of a structure model.
Available since version 8.

GetCalcSupport (_node_number : long) : IRobotNodeSupportData
Function makes available definition of an automatic support at a specified node.
Available since version 8.

Complex attributes of a node

Robot Object Model provides access to structure node attributes by means of a uniform mechanism of labels.

IRobotNodeSupportData

Support parameters are described in the Robot Object Model by means of a set of values defined as RobotNodeSupportData. An object of this type contains data with the labels of the ILT_NODE_SUPPORT type.

Properties:

UX : bool
UY : bool
UZ : bool
RX : bool
RY : bool
RZ : bool
KX : double
KY : double
KZ : double
HX : double
HY : double
HZ : double
Alpha : double
angle value in radians

Beta : double
angle value in radians

Gamma : double
angle value in radians

AX : double
AY : double
AZ : double
BX : double
BY : double
BZ : double
GlobalCoordSystem : bool
definition of a rigid support in the global / local system
Available since version 2.5.

ElasticLinear : bool

linear elasticity distribution; If the flag assumes value different from zero (True), then values of support elastic coefficients will be given per one running meter of length. Value of the elastic coefficient in a node is determined in the following manner:
- for each node, the distance between centers of segments between successive nodes is determined
- the determined distance is multiplied by the coefficient per one running meter defined by the user
- the determined value is defined in the node

Available since version 2.5.

ElasticSurface : bool

surface elasticity distribution; If the flag assumes value different from zero (True), then value of support elasticity is given per surface unit and it is assigned to nodes in proportion to the area of elements adjoining each of the support nodes
Available since version 2.5.

NonlinearModel : IRobotNonlinearLinkMngr

non-linearity support
Available since version 3.

Operations:

IsFixed (direction : IRobotNodeSupportFixingDirection) : bool

The function checks if the given direction is blocked.

SetFixed (direction : IRobotNodeSupportFixingDirection, fixed : bool)

The function allows one to block / un-block an indicated direction.

GetOneDir (direction : IRobotNodeSupportFixingDirection) : IRobotNodeSupportOneDirectionFixingType

The function returns the manner of unilateral blocking of the indicated direction for support.

SetOneDir (direction : IRobotNodeSupportFixingDirection, how_blocked : IRobotNodeSupportOneDirectionFixingType)

The function sets the manner of unilateral blocking of the indicated direction for the support.

SetAdvanced (_type : IRobotAdvancedSupportType, _b : double, _h : double)

Function defines support parameters applied during RC structure analysis.
Available since version 2.5.

GetAdvanced (_b : double*, _h : double*) : IRobotAdvancedSupportType

Function returns support parameters applied during RC structure analysis.
Available since version 2.5.

IRobotNodeSupportFixingDirection

Identifiers of directions that may be blocked out or released in a support.

Properties:

I_NSFD_UX : = 0
I_NSFD_UY : = 1
I_NSFD_UZ : = 2
I_NSFD_RX : = 3
I_NSFD_RY : = 4
I_NSFD_RZ : = 5

IRobotNodeSupportOneDirectionFixingType

Identifiers of the manner of unilateral blocking of the indicated direction for supports.

Properties:

I_NSODFT_NONE : = 0
I_NSODFT_MINUS : = -1
I_NSODFT_PLUS : = 1

IRobotEmitter

Emitter definition.

Properties:

VariableMeshDensityIncrement : bool
Available since version 1.7.

H0 : double
Available since version 1.7.

R1 : double
Available since version 1.7.

R2 : double
Available since version 1.7.

EstimatedElemNumber : long
estimated element number in the sphere (diameter R1)
Available since version 1.7.

IRobotAdvancedSupportType

Available since version 2.5.

Properties:

I_AST_NODAL :
support defined in a node
Available since version 2.5.

I_AST_COLUMN_RECTANGULAR :
rectangular column - it is required to define width b and height h of a column cross-section
Available since version 2.5.

I_AST_COLUMN_CIRCULAR :
round column - it is required to define diameter b of a column cross-section
Available since version 2.5.

I_AST_WALL_MASONRY :
masonry support defined by specifying the wall width b
Available since version 2.5.

I_AST_WALL_CONCRETE :
concrete support defined by specifying the wall width b

Available since version 2.5.

Rigid links

Available since version 2.5.

IRobotNodeRigidLinkData

Rigid link parameters for nodes.
Available since version 2.5.

Properties:

UX : bool
blocked displacement in the UX direction
Available since version 2.5.

UY : bool
blocked displacement in the UY direction
Available since version 2.5.

UZ : bool
blocked displacement in the UZ direction
Available since version 2.5.

RX : bool
blocked displacement in the RX direction
Available since version 2.5.

RY : bool
blocked displacement in the RY direction
Available since version 2.5.

RZ : bool
blocked displacement in the RZ direction
Available since version 2.5.

IRobotNodeRigidLinkDef

Definition of a node rigid link.
Available since version 2.5.

Properties:

Master : long
number of master node
Available since version 2.5.

Slaves : string
selection of slave nodes
Available since version 2.5.

LabelName : string
name of the label storing data for the link
Available since version 2.5.

IRobotNodeRigidLinkServer

Server allowing definition of rigid links for nodes. Moreover, the server provides access to the list of all the rigid links defined.
Available since version 2.5.

Properties:

Count : long
number of defined rigid links
Available since version 2.5.

Operations:

Set (_master : long, _slaves : string, _label_name : string) : bool
Function defines a rigid link for the indicated nodes.
Available since version 2.5.

RemoveSlave (_slave : long)

Function deletes the specified slave node from the definition of the link which includes the node

Available since version 2.5.

FindMaster (_master : long) : long

Function locates a rigid link defined for the specified master node and returns index of the object describing this link.

If no connection has been defined for the indicated node, then zero value is returned.

Available since version 2.5.

Get (_def_idx : long) : IRobotNodeRigidLinkDef

Function returns object defining the selected rigid link. By activating this function with successive indexes from 1 to Count, the user may obtain information about all the rigid links defined.

Available since version 2.5.

Remove (_def_idx : long)

Function deletes the rigid link for which definition parameters are described by the object with the indicated index on the list of rigid links defined.

Available since version 2.5.

GetLabel (_def_idx : long) : IRobotLabel

Function returns label storing parameters of the selected rigid link.

Available since version 2.5.

FindSlave (_slave_node : long) : long

Function locates the rigid link, in which the node specified occurs as a slave node and returns index of the object describing this link. If no link has been defined for the determined node, then zero value is returned.

Available since version 2.5.

Find (_node_num : long) : long

Function locates the rigid link in which the specified node occurs (either as a master node or slave node) and returns index of the object describing this link. If no link has been defined for the determined node, then zero value is returned.

Available since version 2.5.

FindLabel (_label_name : string, _start_def_idx : long = 1) : long

Function locates the rigid link described by the label of the specified name and returns first index of the object describing this link beginning with the determined start index.

Available since version 2.5.

Compatible nodes

Available since version 2.5.

IRobotNodeCompatibilityData

Compatible node.
Available since version 2.5.

Properties:

UX : bool
blocked direction (X axis)
Available since version 2.5.

UY : bool
blocked direction (Y axis)
Available since version 2.5.

UZ : bool
blocked direction (Z axis)
Available since version 2.5.

RX : bool
blocked direction (rotation about X axis)
Available since version 2.5.

RY : bool
blocked direction (rotation about Y axis)
Available since version 2.5.

RZ : bool
blocked direction (rotation about Z axis)
Available since version 2.5.

AX : double
damping coeffcient (X axis)
Available since version 2.5.

AY : double
damping coeffcient (Y axis)
Available since version 2.5.

AZ : double
damping coeffcient (Z axis)
Available since version 2.5.

KX : double
elastic coefficient
Available since version 2.5.

KY : double
elastic coefficient

Available since version 2.5.

KZ : double
elastic coefficient
Available since version 2.5.

Alpha : double
value of the compatibility angle (Z axis)
Available since version 2.5.

Beta : double
value of the compatibility angle (Y axis)
Available since version 2.5.

Gamma : double
value of the compatibility angle (X axis)
Available since version 2.5.

HX : double

Available since version 2.5.

HY : double

Available since version 2.5.

HZ : double

Available since version 2.5.

BX : double

Available since version 2.5.

BY : double

Available since version 2.5.

BZ : double

Available since version 2.5.

NonlinearModel : IRobotNonlinearLinkMngr
non-linearity support
Available since version 3.

IRobotNodeCompatibilityDef

Definition of compatible node.
Available since version 2.5.

Properties:

Main : long
number of master node
Available since version 2.5.

Compatible : long
number of compatible node
Available since version 2.5.

LabelName : string
name of the label storing data for a compatible node
Available since version 2.5.

Bars : String
list of bars for which the compatibility model is defined
Available since version 2.5.

IRobotNodeCompatibilityServer

Server allowing definition of compatible nodes.
Available since version 2.5.

Properties:

Count : long
number of compatible nodes defined
Available since version 2.5.

Operations:

FindMain (_main_node : long, _start_def_idx : long = 1) : long
Function returns index of the first located definition of a compatible node in the specified master node beginning with the start index.
Available since version 2.5.

Set (_node : long, _bar_list : string, _label_name : string) : bool
Function defines a compatible node for the selected bars at the indicated node. The selected bars must converge at the given node.
Available since version 2.5.

Get (_def_idx : long) : IRobotNodeCompatibilityDef
Function returns object containing compatible node parameters. By activating this function with indexes from 1 to Count, the user may obtain information about all the compatible nodes defined.

Available since version 2.5.

Remove (_def_idx : long)

Function deletes the compatible node whose definition method describes the object of the specified index on the list of all compatible nodes.

Available since version 2.5.

GetLabel (_def_idx : long) : IRobotLabel
Function returns label describing parameters of the determined compatible node.
Available since version 2.5.

FindCompatible (_compatible _node : long) : long
Function returns definition index of the compatible node of the indicated user number.
Available since version 2.5.

Find (_node : long, _start_def_idx : long = 1) : long
Function returns index of the first located compatible node definition in which the specified node occurs either as a master node or compatible node.
Available since version 2.5.

FindLabel (_label_name : string, _start_def_idx : long = 1) : long
Function returns index of the first located compatible node definition with the specified name.
Available since version 2.5.

Bars

IRobotBar

Derived from IRobotDataObject

Each bar in a structure is represented in the Object Model by a data object of the type: I_OT_BAR. The data and functionality related to the bar are available through time history RobotBar interface.
The following complex bar attributes are defined by means of the labelling mechanism:

Attribute	Label type
Section	I_LT_BAR_SECTION
Release	I_LT_BAR_RELEASE
Material	I_LT_BAR_MATERIAL
Offset	I_LT_BAR_OFFSET

Properties:

StartNode : long
number of the initial bar node

EndNode : long
number of the end bar node

Gamma : double
Gamma angle value defined for the bar. The gamma angle may be defined both for a bar and for a section. The resultant angle is a sum of both values. The value describes the angle for the bar.

Start : IRobotBarEnd
bar beginning

End : IRobotBarEnd
bar end

UniqueId : long
unique identifier of structure component
Available since version 1.7.

TensionCompression : IRobotBarTensionCompression
manner in which the bar carries stresses
Available since version 2.5.

ShearForces : bool
flag indicating if shearing forces acting on a bar will be included in calculations of structure deformation
Available since version 2.5.

Length : double
bar length
Available since version 2.5.

ReversedSection : bool
flag indicating if section assigned to the bar is inverted (section beginning is applied to the bar end)
Available since version 2.5.

TrussBar : bool
flag indicating if the bar is a truss bar
Available since version 2.5.

Elements : IRobotCollection
collection containing one or more bar elements defined for the bar (the collection components are objects of the IRobotBarElement type)
Available since version 3.

IsSuperBar : bool
flag indicating that the bar is a superbar
Available since version 3.

ReversedOffset : bool
flag indicating if offset assigned to a bar is reversed with respect to bar orientation
Available since version 3.5.

InactiveBar : bool
flag enabling exclusion of a bar from the structure model - making it "inactive"
Available since version 4.5.

Name : string
bar name
Available since version 7.5.

NameTemplate : string
bar name pattern
Available since version 7.5.

Operations:

SetSection (_section_name : string, _reverse : bool)
Function assigns section of the indicated name to the bar, just as the SetLabel function does. Moreover, it allows inverting the section - applying section end to the bar beginning.
Available since version 2.5.

GetSimpleBars () : IRobotSelection
Function returns the selection of simple bars which are the superbar components. Prior to activating the function, the IsSuperBar flag should be checked. If the function is activated for a simple bar, it causes a critical error.
Available since version 3.

SetOffset (_offset_name : string, _reverse : bool)
Function assigns offset with a specified name to a bar, identically as the SetLabel function. In addition, it enables reversing the offset.
Available since version 3.5.

ChangeOrientation ()
Function changes bar orientation to opposite.
Available since version 4.5.

SetOrientation (_dir : IRobotGeoCoordinateAxisSense)
Function adjusts bar orientation to the defined sense of the coordinate system axis.
Available since version 4.5.

GetLocalMtx (_m00 : double*, _m01 : double*, _m02 : double*, _m10 : double*, _m11 : double*, _m12 : double*, _m20 : double*, _m21 : double*, _m22 : double*)
Function returns transformation matrix for the bar local system.
Available since version 5.

CalcGamma (_local_Z_x : double, _local_Z_y : double, _local_Z_z : double) : double
Function calculates a Gamma angle value for a bar based on the coordinates (in the global system) of the local Z direction vector.

IRobotBarServer

Derived from IRobotDataObjectServer
Server of bars manages the bars defined in the structure

Properties:

FreeNumber : long
first free user number available for bars
Available since version 1.7.

NonlinearHingeModels : IRobotNonlinearHingeModelServer
server of non-linear hinge models
Available since version 3.

NonlinearHinges : IRobotNonlinearHingeServer
Non-linear hinge server
Available since version 3.

Operations:

Create (bar_num : long, start_node : long, end_node : long)
The function creates between the indicated nodes a bar with the defined number.

FindWithId (unique_id : long) : long
Function returns a number of the bar with a specified unique identifier. If such a bar is not found, then, zero value is returned.
Available since version 1.7.

SetTensionCompression (_bar_sel : IRobotSelection, _tc : IRobotBarTensionCompression)
Function defines manner of carrying stresses for the specified bar selection.
Available since version 2.5.

SetShearForces (_bar_sel : IRobotSelection, _set : bool = -1)
For the indicated selection of bars, the function sets flag indicating if shearing forces are taken into account for them.
Available since version 2.5.

CreateSuperBar (_number : long, _simple_bars : string)
Function generates a superbar with the user-defined number, composed of the indicated simple bars.
Available since version 3.

Update ()

Function updates information on mutual relations between bars and nodes in a strucuture. It should be activated after generation of all structure bars and before definition of bar-node properties.
Available since version 3.5.

ChangeOrientation (_bars : string)
Function changes orientation of selected bars to opposite
Available since version 4.5.

SetOrientation (_bars : string, _dir : IRobotGeoCoordinateAxisSense)
Function adjusts orientation of selected bars to the defined sense of the coordinate system axis.
Available since version 4.5.

IsInactive (_bar_num : long) : bool
Function checking if a given bar is assigned the "inactive" status.
Available since version 4.5.

SetInactive (_bars : string, _inactive : bool = true)
Funtion setting the "inactive" status for the defined bar list.
Available since version 4.5.

GetLocalMtx (_bar_num : long, _m00 : double*, _m01 : double*, _m02 : double*, _m10 : double*, _m11 : double*, _m12 : double*, _m20 : double*, _m21 : double*, _m22 : double*)
Function returns transformation matrix for the bar local system.
Available since version 5.

CalcGamma (_bar_num : long, _local_Z_x : double, _local_Z_y : double, _local_Z_z : double) : double
Function calculates a Gamma angle value for a bar based on the coordinates (in the global system) of the local Z direction vector.

SetTrussBar (_bars : string, _truss_bar : bool)
Function enables definition of truss bars.

IsTrussBar (_bar_num : long) : bool
Function returns a value different from zero (True) if a given bar is a truss bar.

GetName (_bar_num : long) : string
Function returns a name of the specified bar.
Available since version 7.5.

GetNameTemplate (_bar_num : long) : string
Function returns a name pattern of the specified bar.
Available since version 7.5.

SetNameTemplate (_bar_num : long, _name_tmpl : string)
Function sets a name pattern for the specified bar.
Available since version 7.5.

IRobotBarEnd

One can indicate a beginning and an end of each bar. The object of RobotBarEnd type, describing one of the bar ends, is created to make it possible to carry out operations affecting only the beginning or end of a bar (and not the entire bar).

Properties:

Node : long
node number (user-defined)

Operations:

SetLabel (lab_type : IRobotLabelType, lab_name : string)
The function applies a label of a given type to the object.

HasLabel (lab_type : IRobotLabelType) : bool
The function returns a non-zero value (True) if a label of the given type has been defined for the object.

GetLabel (lab_type : IRobotLabelType) : IRobotLabel
The function returns the label of the given type defined for the object.

RemoveLabel (lab_type : IRobotLabelType)
The function removes the label of the given type from the object.

GetLabelName (lab_type : IRobotLabelType) : string
The function returns the label of the given type, applied to the object.

GetLabels () : IRobotCollection
The function returns a collection containing all labels applied to the object.

GetOffsetValue () : IRobotBarEndOffsetData
Function calculates and returns total values of offset at a bar end.
Available since version 3.5.

IRobotBarTensionCompression

Available since version 2.5.

Properties:

I_BTC_STANDARD : = 0
the bar carries both compressive and tensile stresses
Available since version 2.5.

I_BTC_TENSION_ONLY : = 1
the bar carries only tensile stresses

Available since version 2.5.

I_BTC_COMPRESSION_ONLY : = 2
the bar carries only compressive stresses
Available since version 2.5.

IRobotBarElement

Bar element
Available since version 3.

Properties:

Number : long

Available since version 3.

StartNode : long
Number of origin node
Available since version 3.

EndNode : long
Number of end node
Available since version 3.

IsCalc : bool
flag indicating if the element has been created automatically during generation of the structure calculation model
Available since version 3.

Inactive : bool
flag enabling exclusion of a bar element from the structure calculation model
Available since version 4.5.

Complex attributes of a bar

Complex attributes of a bar are defined by means of the labelling mechanism.

IRobotBarOffsetData

The structure describing an offset defined for a bar.

Properties:

CoordinateSystem : IRobotGeoCoordinateSystem
the coordinate system in which the offsets at the bar ends are defined

Start : IRobotBarEndOffsetData

offset at the beginning of a bar

End : IRobotBarEndOffsetData
offset at the end of a bar

Position : IRobotBarOffsetAutoPosition
position with respect to bar section (in local coordinate system)
Available since version 3.5.

AxisOffset : IRobotBarOffsetAutoPosition
ObjectNumber : long

IRobotBarEndOffsetData

The structure describing a shift at one of bar ends.

Properties:

UX : double
UY : double
UZ : double
MemberLength : IRobotBarOffsetMemberLength

IRobotBarEndBracketType

Available types of brackets.

Properties:

I_BEBT_PLATES : = 1
I_BEBT_SECTION : = 2

IRobotBarEndBracketDataValue

A set of identifiers has been defined that determines particular parameters (attributes), describing a bracket at bar end. The identifier is used as a parameter of the function that gets or sets the value of the relevant bracket attribute.

Properties:

I_BEBDV_LENGTH : = 1
I_BEBDV_HEIGHT : = 2
I_BEBDV_WIDTH : = 3
I_BEBDV_THICKNESS_1 : = 4
I_BEBDV_THICKNESS_2 : = 5

IRobotBarEndBracketData

The structure describing a bracket at bar end.

Properties:

Type : IRobotBarEndBracketType
bracket type

Operations:

SetValue (param : IRobotBarEndBracketDataValue, value : double, relative : bool)
The function sets the value of the indicated bracket parameter.

GetValue (param : IRobotBarEndBracketDataValue) : double
The function returns the value of the indicated bracket parameter. In order to find out if the returned value is of absolute or relative type, one should call the IsValueRelative function, providing the same identifier of bracket attribute.

IsValueRelative (param : IRobotBarEndBracketDataValue) : bool
The function returns information about the character of the given bracket parameter. If zero is returned (False), it means that the indicated parameter is determined in absolute quantities, while a non-zero value (True) means that the bracket parameter is determined relative to the parameters of the bar section.

IRobotBarOffsetAutoPosition

Available predefined position of an offset with respect to section.
Available since version 3.5.

Properties:

I_BOAP_0_0 : = 0

Available since version 3.5.

I_BOAP_VPY_VPZ : = 1

Available since version 3.5.

I_BOAP_0_VPZ : = 2

Available since version 3.5.

I_BOAP_VY_VPZ : = 3

Available since version 3.5.

I_BOAP_VPY_0 : = 4

Available since version 3.5.

I_BOAP_VY_0 : = 5

Available since version 3.5.

I_BOAP_VPY_VZ : = 6

Available since version 3.5.

I_BOAP_0_VZ : = 7

Available since version 3.5.

I_BOAP_VY_VZ : = 8

Available since version 3.5.

IRobotBarOffsetMemberLength

Properties:

I_BOML_IN_THE_AXIS : = 0
I_BOML_INCREASE_LENGTH : = 1
I_BOML_REDUCE_LENGTH : = 2

Sections

Available since version 2.5.

IRobotBarSectionShapeType

A set of section shape types recognized by the program.

Properties:

I_BSST_UNKNOWN : = 0
I_BSST_CAE : = 1
I_BSST_CAEP : = 2
I_BSST_CAI : = 3
I_BSST_CAIP : = 4
I_BSST_DCEC : = 5
I_BSST_DCED : = 6
I_BSST_DCEP : = 7
I_BSST_DCIG : = 8
I_BSST_DCIP : = 9
I_BSST_HEA : = 10
I_BSST_HEAA : = 11
I_BSST_HEB : = 12
I_BSST_HEC : = 13
I_BSST_HEM : = 14
I_BSST_HER : = 15
I_BSST_HHEA : = 16

I_BSST_HHEB : = 17
I_BSST_HHEM : = 18
I_BSST_IIPE : = 19
I_BSST_IPE : = 20
I_BSST_IPEA : = 21
I_BSST_IPEO : = 22
I_BSST_IPER : = 23
I_BSST_IPEV : = 24
I_BSST_IPN : = 25
I_BSST_MHEA : = 26
I_BSST_MHEB : = 27
I_BSST_MHEM : = 28
I_BSST_MIPE : = 29
I_BSST_PRS : = 30
I_BSST_TCAR : = 31
I_BSST_TEAE : = 32
I_BSST_TEAI : = 33
I_BSST_THEX : = 34
I_BSST_TREC : = 35
I_BSST_TRON : = 36
I_BSST_UAP : = 37
I_BSST_UPN : = 38
I_BSST_UUAP : = 39
I_BSST_UUPN : = 40
I_BSST_FRTG : = 41
I_BSST_UPAF : = 42
I_BSST_BOX : = 91
I_BSST_RECT : = 92
I_BSST_TUBE : = 93
I_BSST_ISYM : = 94
I_BSST_INSYM : = 95
I_BSST_TUSER : = 96
I_BSST_CUSER : = 97
I_BSST_TBETC : = 98
I_BSST_WELD_CROSS : = 201
I_BSST_DRECT : = 99
Available since version 1.7.

I_BSST_COLD_SIGMA1 : = 1001
Available since version 1.7.

I_BSST_COLD_SIGMA2 : = 1002
Available since version 1.7.

I_BSST_COLD_ZED1 : = 1003
Available since version 1.7.

I_BSST_COLD_U : = 1004
Available since version 1.7.

I_BSST_COLD_CE1 : = 1005

Available since version 1.7.

I_BSST_COLD_ANGL : = 1006
Available since version 1.7.

I_BSST_COLD_OMEGA : = 1007
Available since version 1.7.

I_BSST_COLD_SO1 : = 1008
Available since version 1.7.

I_BSST_COLD_RIVE1 : = 1009
Available since version 1.7.

I_BSST_USER_BOX : = 91

Available since version 2.5.

I_BSST_USER_RECT : = 92

Available since version 2.5.

I_BSST_USER_TUBE : = 93

Available since version 2.5.

I_BSST_USER_I_BISYM : = 94

Available since version 2.5.

I_BSST_USER_I_MONOSYM : = 95

Available since version 2.5.

I_BSST_USER_T_SHAPE : = 96

Available since version 2.5.

I_BSST_USER_C_SHAPE : = 97

Available since version 2.5.

I_BSST_USER_CROSS : = 201

Available since version 2.5.

I_BSST_WOOD_RECT : = 41
timber rectangular section
Available since version 2.5.

I_BSST_WOOD_DRECT : = 99

timber double rectangular section
Available since version 2.5.

I_BSST_RECT_FILLED : = 43
steel rectangular solid section
Available since version 2.5.

I_BSST_CIRC_FILLED : = 44
steel round solid section
Available since version 2.5.

I_BSST_CONCR_COL_R : = -108

Available since version 2.5.

I_BSST_CONCR_COL_T : = -107

Available since version 2.5.

I_BSST_CONCR_COL_L : = -106

Available since version 2.5.

I_BSST_CONCR_COL_Z : = -105

Available since version 2.5.

I_BSST_CONCR_COL_P : = -104

Available since version 2.5.

I_BSST_CONCR_COL_C : = -103

Available since version 2.5.

I_BSST_CONCR_COL_CH : = -102

Available since version 2.5.

I_BSST_CONCR_COL_CQ : = -101

Available since version 2.5.

I_BSST_CONCR_BEAM_RECT : = -3
RC rectangular beam
Available since version 2.5.

I_BSST_CONCR_BEAM_T : = -2

Available since version 2.5.

I_BSST_CONCR_BEAM : = -1
RC beam of any shape
Available since version 2.5.

I_BSST_COMP_2C_FACE : = 1101
two C-sections face to face
Available since version 4.

I_BSST_COMP_2C_BACK : = 1102
two C-sections back to back
Available since version 4.

I_BSST_COMP_2I : = 1103
two I-sections
Available since version 4.

I_BSST_COMP_CI : = 1104
C-section and I-section
Available since version 4.

I_BSST_COMP_2LI : = 1105
two angles and one I-section
Available since version 4.

I_BSST_COMP_4L_FACE : = 1106
four angles face to face
Available since version 4.

I_BSST_COMP_4L_BACK : = 1107
four angles with legs back to back
Available since version 4.

I_BSST_COMP_2L_SHORT : = 1108
two angles with shorter legs back to back
Available since version 4.

I_BSST_COMP_2L_LONG : = 1109
two angles with longer legs back to back
Available since version 4.

I_BSST_COMP_2L_CROSS : = 1110
two angles - cross shape
Available since version 4.

I_BSST_USER_BOX_2 : = 102
I_BSST_CCL : = 45
semi-closed C-section

I_BSST_URND : = 46
rounded C-section - rotated

I_BSST_TRND : = 47
rectangular pipe

I_BSST_CUAP : = 48
C-sections set face to face, welded

I_BSST_WOOD_CIRC : = 100
round timber section - solid

I_BSST_USER_CIRC_FILLED : = 101
user steel section - round, solid

I_BSST_USER_POLYGONAL : = 103
pipe section - shaped like a regular polygon

I_BSST_COLD_C_PLUS : = 1010
cold-formed C-section - semi-closed, with bends

I_BSST_COLD_SIGMA_SL : = 1011
cold-formed Sigma SL section

I_BSST_COLD_SIGMA : = 1012
cold-formed Sigma section

I_BSST_COLD_Z : = 1013
cold-formed Z-section

I_BSST_COLD_L_LIPS : = 1014
cold-formed C-section - semi-closed

I_BSST_COLD_Z_ROT : = 1015
cold-formed Z-section in the local - central coordinate system

I_BSST_COMP_2L_FACE_SHORT : = 1111
two angles - shorter legs

I_BSST_COMP_2L_FACE_LONG : = 1112
two angles - longer legs

I_BSST_COMP_CI_BACK : = 1113
C-section and I-section set back to back

I_BSST_COMP_2C_FACE_WELD : = 1201
two C-sections set face to face, welded

I_BSST_COMP_2C_BACK_WELD : = 1202
two C-sections set back to back, welded

I_BSST_COMP_2I_WELD : = 1203
two welded I-sections

I_BSST_COMP_CI_WELD : = 1204
C-section and I-section set face to face - welded

I_BSST_COMP_2LI_WELD : = 1205
two angles and I-section - welded

I_BSST_COMP_4L_FACE_WELD : = 1206
four angles with legs set face to face - welded

I_BSST_COMP_4L_BACK_WELD : = 1207
four angles set in the form of a cross - welded

I_BSST_COMP_2L_SHORT_WELD : = 1208
two angles - shorter legs, welded

I_BSST_COMP_2L_LONG_WELD : = 1209
two angles - longer legs, welded

I_BSST_COMP_2L_CROSS_WELD : = 1210
two angles set in the form of a cross - welded

I_BSST_COMP_2L_FACE_SHORT_WELD : = 1211
I_BSST_COMP_2L_FACE_LONG_WELD : = 1212
I_BSST_COMP_CI_BACK_WELD : = 1213

IRobotBarSectionDataValue

A set of identifiers is defined to determine particular parameters (attributes) describing a bar section. Such an identifier is provided as the parameter of a function that gets or sets the value of the relevant section attribute.

Properties:

I_BSDV_AX : = 0
I_BSDV_AY : = 1
I_BSDV_AZ : = 2
I_BSDV_IX : = 3
I_BSDV_IY : = 4
I_BSDV_IZ : = 5
I_BSDV_VY : = 6
I_BSDV_VPY : = 7
I_BSDV_VZ : = 8
I_BSDV_VPZ : = 9
I_BSDV_SURFACE : = 10
painting area (perimeter)

I_BSDV_WEIGHT : = 11
nominal weight per length unit

I_BSDV_D : = 12
section height - basic (maximum) vertical dimension of a section

I_BSDV_BF : = 13
section width - basic (maximum) horizontal dimension of a section

I_BSDV_TW : = 14
web thickness / vertical wall thickness

I_BSDV_TF : = 15
flange thickness / horizontal wall thickness

I_BSDV_RA : = 16
fillet radius

I_BSDV_RI : = 17
fillet radius

I_BSDV_S : = 18
spacing - distance between section elements (double angles)

I_BSDV_ZY : = 19
section plasticity modulus - bending around Y axis

I_BSDV_ZZ : = 20
section plasticity modulus - bending around Z axis

I_BSDV_WX : = 21
section modulus for calculation of torsional stresses

I_BSDV_WY : = 22
section modulus for calculation of limit shear stresses along Y axis

I_BSDV_WZ : = 23
section modulus for calculation of limit shear stresses along Z axis

I_BSDV_GAMMA : = 24
angle between principal and main coordinate system axes

I_BSDV_IOMEGA : = 25
first moment of area

I_BSDV_P1_LENGTH : = 26
length of plate 1 in the CROSS section type

I_BSDV_P1_THICKNESS : = 27
thickness of plate 1 in the CROSS section type

I_BSDV_P2_LENGTH : = 28
length of plate 2 in the CROSS section type

I_BSDV_P2_THICKNESS : = 29
thickness of plate 2 in the CROSS section type

I_BSDV_P3_LENGTH : = 30
length of plate 3 in the CROSS section type

I_BSDV_P3_THICKNESS : = 31
thickness of plate 3 in the CROSS section type

I_BSDV_P4_LENGTH : = 32
length of plate 4 in the CROSS section type

I_BSDV_P4_THICKNESS : = 33
thickness of plate 4 in the CROSS section type

I_BSDV_BF2 : = 34
second section width (compare I_BSDV_BF)

I_BSDV_TF2 : = 35
second flange thickness (compare I_BSDV_TF)

I_BSDV_DIM1 : = 36
first dimension

I_BSDV_DIM2 : = 37
second dimension

I_BSDV_DIM3 : = 38
third dimension

I_BSDV_ANGLE1 : = 39
additional angle

I_BSDV_ANGLE2 : = 40
additional angle

IRobotBarSectionType

The set of section types recognized by the program has been defined.

Properties:

I_BST_STANDARD : = 0
standard section (no additional list of data sets describing a tapered section); this section type is used for all sections except for non-standard user tapered sections and compound sections.

I_BST_NS_BOX : = 1

I_BST_NS_I : = 2
I_BST_NS_II : = 3
I_BST_NS_TUBE : = 4
I_BST_NS_RECT : = 5
I_BST_NS_C : = 6
I_BST_NS_L : = 7
I_BST_NS_LP : = 8
I_BST_NS_Z : = 9
I_BST_NS_ZP : = 10
I_BST_NS_T : = 11
I_BST_NS_H : = 12
I_BST_NS_XT : = 13
I_BST_NS_XI : = 14
I_BST_NS_DRECT : = 15
I_BST_NS_CROSS : = 16
I_BST_NS_HOLE : = 99
I_BST_COMPLEX : = 1000
compound section
Available since version 4.

I_BST_NS_BOX_2 : = 17
I_BST_NS_POLYGONAL : = 18

IRobotBarSectionNonstdDataValue

A set of identifiers describing section parameters is defined for each type of non-standard user-defined section.

Properties:

I_BSNDV_BOX_H : = 0
I_BSNDV_BOX_B : = 1
I_BSNDV_BOX_TF : = 2
I_BSNDV_BOX_TW : = 3
I_BSNDV_I_H : = 0
I_BSNDV_I_B : = 1
I_BSNDV_I_TF : = 2
I_BSNDV_I_TW : = 3
I_BSNDV_II_H : = 0
I_BSNDV_II_TW : = 1
I_BSNDV_II_TF1 : = 2
I_BSNDV_II_TF2 : = 3
I_BSNDV_II_B1 : = 4
I_BSNDV_II_B2 : = 5
I_BSNDV_TUBE_D : = 0
I_BSNDV_TUBE_T : = 1
I_BSNDV_RECT_B : = 0
I_BSNDV_RECT_H : = 1
I_BSNDV_RECT_T : = 2
I_BSNDV_C_H : = 0

I_BSNDV_C_B : = 1
I_BSNDV_C_TF : = 2
I_BSNDV_C_TW : = 3
I_BSNDV_L_H : = 0
I_BSNDV_L_B : = 1
I_BSNDV_L_TF : = 2
I_BSNDV_L_TW : = 3
I_BSNDV_Z_H : = 0
I_BSNDV_Z_B : = 1
I_BSNDV_Z_TF : = 2
I_BSNDV_Z_TW : = 3
I_BSNDV_T_H : = 0
I_BSNDV_T_B : = 1
I_BSNDV_T_TF : = 2
I_BSNDV_T_TW : = 3
I_BSNDV_H_H : = 0
I_BSNDV_H_B : = 1
I_BSNDV_H_B1 : = 2
I_BSNDV_H_TF : = 3
I_BSNDV_H_TW : = 4
I_BSNDV_XT_H : = 0
I_BSNDV_XT_H1 : = 1
I_BSNDV_XT_B : = 2
I_BSNDV_XT_TF : = 3
I_BSNDV_XT_TW : = 4
I_BSNDV_XI_H : = 0
I_BSNDV_XI_H1 : = 1
I_BSNDV_XI_B : = 2
I_BSNDV_XI_TF : = 3
I_BSNDV_XI_TW : = 4
I_BSNDV_DRECT_B : = 0
I_BSNDV_DRECT_H : = 1
I_BSNDV_DRECT_D : = 2
I_BSNDV_CROSS_P1_L : = 0
I_BSNDV_CROSS_P2_L : = 1
I_BSNDV_CROSS_P3_L : = 2
I_BSNDV_CROSS_P4_L : = 3
I_BSNDV_CROSS_P1_T : = 4
I_BSNDV_CROSS_P2_T : = 5
I_BSNDV_CROSS_P3_T : = 6
I_BSNDV_CROSS_P4_T : = 7
I_BSNDV_HOLE_B : = 0
I_BSNDV_HOLE_H : = 1
I_BSNDV_HOLE_X : = 2
I_BSNDV_HOLE_Z : = 3
I_BSNDV_BOX_2_H : = 0
I_BSNDV_BOX_2_B : = 1
I_BSNDV_BOX_2_B1 : = 2
I_BSNDV_BOX_2_TF : = 3
I_BSNDV_BOX_2_TW : = 4
I_BSNDV_POLYGONAL_N : = 0

I_BSNDV_POLYGONAL_D : = 1
I_BSNDV_POLYGONAL_T : = 2

IRobotBarSectionConcreteDataValue

Characteristic parameters for RC sections.
Available since version 2.5.

Properties:

I_BSCDV_COL_H : = 0

Available since version 2.5.

I_BSCDV_COL_B : = 1

Available since version 2.5.

I_BSCDV_COL_H1 : = 2

Available since version 2.5.

I_BSCDV_COL_L1 : = 3

Available since version 2.5.

I_BSCDV_COL_H2 : = 4

Available since version 2.5.

I_BSCDV_COL_L2 : = 5

Available since version 2.5.

I_BSCDV_COL_DE : = 6

Available since version 2.5.

I_BSCDV_COL_N : = 7

Available since version 2.5.

I_BSCDV_BEAM_H : = 0

Available since version 2.5.

I_BSCDV_BEAM_B : = 1

Available since version 2.5.

I_BSCDV_BEAM_PL : = 2

Available since version 2.5.

I_BSCDV_BEAM_RL : = 3

Available since version 2.5.

I_BSCDV_BEAM_EL1 : = 4

Available since version 2.5.

I_BSCDV_BEAM_EL2 : = 5

Available since version 2.5.

I_BSCDV_BEAM_PR : = 6

Available since version 2.5.

I_BSCDV_BEAM_RR : = 7

Available since version 2.5.

I_BSCDV_BEAM_ER1 : = 8

Available since version 2.5.

I_BSCDV_BEAM_ER2 : = 9

Available since version 2.5.

I_BSCDV_BEAM_HG1 : = 10

Available since version 2.5.

I_BSCDV_BEAM_BG1 : = 11

Available since version 2.5.

I_BSCDV_BEAM_HG2 : = 12

Available since version 2.5.

I_BSCDV_BEAM_BG2 : = 13

Available since version 2.5.

I_BSCDV_BEAM_T_H : = 0

Available since version 2.5.

I_BSCDV_BEAM_T_B : = 1

Available since version 2.5.

I_BSCDV_BEAM_T_HF : = 100

Available since version 2.5.

I_BSCDV_BEAM_T_BF : = 101

Available since version 2.5.

I_BSCDV_BEAM_RECT_H : = 0

Available since version 2.5.

I_BSCDV_BEAM_RECT_B : = 1

Available since version 2.5.

IRobotBarSectionConcreteCutsPosition

Position of cuts on an RC section.
Available since version 2.5.

Properties:

I_BSCCP_NONE : = 0
no cuts
Available since version 2.5.

I_BSCCP_LEFT : = 1
cuts on the left side
Available since version 2.5.

I_BSCCP_RIGHT : = 2
cuts on the right side
Available since version 2.5.

IRobotBarSectionData

Data defining a bar section. Object of the type is a Data component of each label of the ILT_BAR_SECTION type.

Properties:

Type : IRobotBarSectionType
section type

ShapeType : IRobotBarSectionShapeType
section shape type

Name : string
section name

NonstdCount : long
number of records describing parameters of non-standard tapered sections (for each section of the I_BST_STANDARD type, the value equals 0)

IsConcrete : bool
flag indicating if it is an RC section that is dealt with
Available since version 2.5.

Concrete : IRobotBarSectionConcreteData
RC section parameters
Available since version 2.5.

MaterialName : string
name of the material associated with a bar section (material type must be consistent with the section type)
Available since version 3.

ElasticParams : IRobotBarSectionElasticParams
elasto-plastic properties
Available since version 3.5.

Members : IRobotBarSectionComplexData
component enabling definition of a compound section;
This component is active after previous setting the I_BST_COMPLEX value for the Type field (section type) and for the ShapeType field (shape type) - an appropriate shape available to compound sections.
Available since version 4.

AliasCount : long
number of additional section names
Available since version 6.1.

Operations:

LoadFromDBase (section_name : string) : bool
The function browses databases for all available sections of the indicated name and reads the data of the found section. If the section of the given name is found and properly read, the function returns the value True (non-zero value); otherwise, it returns False (zero).

GetValue (attribute : IRobotBarSectionDataValue) : double

The function returns the value of the indicated section parameter. Each parameter describing a section has got its identifier. The set of possible identifiers is defined by the RobotBarSectionDataValue type.

SetValue (attribute : IRobotBarSectionDataValue, value : double)

The function sets the value of the indicated section parameter. Each parameter describing a section has got its identifier. The set of possible identifiers is defined by the RobotBarSectionDataValue type.

CreateNonstd (relative_pos : double) : IRobotBarSectionNonstdData

The function creates and returns an object (data set) describing the section in a point indicated relatively along the bar (0 means bar beginning, 1 - bar end). The newly-created data structure is added at the end of objects describing a tapered section. The function is used only in the case of non-standard tapered sections.

GetNonstd (index : long) : IRobotBarSectionNonstdData

A definition of a tapered section consists of a list of objects describing a section in particular points of variability. An index from the range (1 to number of items in the list) is ascribed to each object from the list. The function returns an object with the indicated index.

FindNonstd (relative_pos : double) : long

The function finds an object describing the cross-section at a bar point indicated relative to its length. It returns the index of the object or zero, when the object is not found.

RemoveNonstd (index : long)

The function removes the positions with the indicated index from the list of objects (data sets) describing a tapered section.

CalcNonstdGeometry ()

The function calculates all the parameters determining section geometry on the basis of the non-standard parameters defined by the user. It should be run for a non-standard tapered section after defining all data sets describing sections in the points of variability.

GetAlias (_idx : long) : string

The function returns an additional section name.
Available since version 6.1.

DrawSymbol (_window_handle : long, _left : long, _top : long, _right : long, _bottom : long)

Function draws a graphical symbol of a section in the specified area of the window.
Available since version 7.5.

LoadFromDBase2 (_section_name : string, _dbase_name : string) : bool
Function searches a section of the specified name in the indicated section database, and next reads data of the section found. If a section of the specified name is found and correctly read, function returns the True value (different from zero), otherwise, it returns the False value (zero).
Available since version 7.5.

GetAliasEx (_idx : long, _alias_id : string*) : string
Function returns an additional section name and its identifier.
Available since version 7.5.

FindAlias (_alias_id : string, _alias : string*) : bool
Function searches and returns the additional section name based on a given identifier.
Available since version 7.5.

IRobotBarSectionNonstdData

Set of data used for describing the cross-section of a non-standard tapered section.

Properties:

Position : double
relative position of a cross-section along a bar where a tapered section has been applied (0 - bar beginning, 1 - bar end)

Operations:

SetValue (attribute : IRobotBarSectionNonstdDataValue, value : double)
The function sets the value of the indicated parameter

GetValue (attribute : IRobotBarSectionNonstdDataValue) : double
The function returns tha value of the indicated section parameter.

IRobotBarSectionConcreteData

Characteristic parameters for an RC member section.
Available since version 2.5.

Properties:

IsColumn : bool

Available since version 2.5.

IsBeam : bool

Available since version 2.5.

BeamCutsPosition : IRobotBarSectionConcreteCutsPosition
position of cuts on a beam
Available since version 2.5.

Operations:

SetValue (_param : IRobotBarSectionConcreteDataValue, _val : double)
Function sets the value of the selected section parameter.
Available since version 2.5.

GetValue (_param : IRobotBarSectionConcreteDataValue) : double
Function returns the value of the selected section parameter.
Available since version 2.5.

SetTapered (_h2 : double)
Function defines variable section width for an RC beam.
Available since version 2.5.

GetTapered (_h2 : double*) : bool
Function returns value different from zero (True), if a given section describes an RC beam with variable section width. At the same time, value of the variable section width is entered into the specified parameter. If it is not a section of variable width, then the function returns zero value (False).
Available since version 2.5.

CalcGeometry ()
Function calculates section geometry based on the indicated characteristic values.
Available since version 2.5.

SetReduction (_set : bool, _Ix_cf : double, _Iy_cf : double, _Iz_cf : double)
Function sets coefficients of reduction of the moment of inertia.
Available since version 8.

GetReduction (_Ix_cf : double*, _Iy_cf : double*, _Iz_cf : double*) : bool
If coefficients of reduction of the moment of inertia were set earlier, function returns True value and specifies values of coefficients. Otherwise function returns False value.
Available since version 8.

IRobotBarSectionElasticParams

Elasto-plastic properties for sections.
Available since version 3.5.

Properties:

Active : bool
flag switching on/off definition of elasto-plastic properties
Available since version 3.5.

MaterialModel : IRobotMaterialElasticModel
material model
Available since version 3.5.

N : double

Available since version 3.5.

N1 : double

Available since version 3.5.

N2 : double

Available since version 3.5.

L1 : double

Available since version 3.5.

L2 : double

Available since version 3.5.

IRobotBarSectionComplexData

Definition of compound section.
Available since version 4.

Properties:

Count : long
number of different simple sections composing a compound section;
This number depends on the selected compound section shape - the ShapeType element of the IRobotBarSectionData interface.
Available since version 4.

D : double
spacing d
Available since version 4.

B : double
spacing b
Available since version 4.

Operations:

GetShape (_idx : long) : IRobotBarSectionComponentShape
Function returns the shape identifier for a simple section being a component of a compound section.
Available since version 4.

Get (_shape : IRobotBarSectionComponentShape, _database_name : string*, _section_name : string*) : bool
Function returns section database name and section name for the indicated shape identifier.
Available since version 4.

Set (_shape : IRobotBarSectionComponentShape, _database_name : string, _section_name : string) : bool
Function sets section database name and section name for the indicated shape identifier.
Available since version 4.

GetValue (_shape : IRobotBarSectionComponentShape, _val_id : IRobotBarSectionDataValue) : double
Function returns value of the selected property for the indicated simple section.
Available since version 4.

IRobotBarSectionComponentShape

Shape types of sections being components of a compound section.
Available since version 4.

Properties:

I_BSCS_C : = 1
C-section
Available since version 4.

I_BSCS_I : = 2
I-section
Available since version 4.

I_BSCS_L : = 3
angle
Available since version 4.

I_BSCS_UNDEFINED : = 0

Available since version 4.

Cables

Available since version 2.5.

IRobotBarCableData

Definition of cable parameters.
Available since version 2.5.

Properties:

SectionAX : double
cable section area
Available since version 2.5.

MaterialName : BSTR
name of the cable material
Available since version 2.5.

AssemblingParam : IRobotBarCableAssemblingParamType
assembling parameter
Available since version 2.5.

AssemblingParamValue : double
value of the assembling parameter
Available since version 2.5.

IRobotBarCableAssemblingParamType

Available types of assembling parameters.
Available since version 2.5.

Properties:

I_BCAPT_STRESSES_SIGMA : = 1
definition of normal stress as a cable parameter for the assembling load case
Available since version 2.5.

I_BCAPT_FORCE_FO : = 2
definition of tension force as a cable parameter for the assembling load case

Available since version 2.5.

I_BCAPT_LENGTH_L : = 3
definition of the unloaded cable length as a cable parameter for the assembling load case

Available since version 2.5.

I_BCAPT_ELONGATION_DL : = 5
definition of dilatation as a cable parameter for the assembling load case

Available since version 2.5.

I_BCAPT_ELONGATION_DL_RELATIVE : = 4
definition of relative dilatation as a cable parameter for the assembling load case
Available since version 2.5.

Elastic ground

Available since version 2.5.

IRobotBarElasticGroundData

Elastic ground parameters for bars.
Available since version 2.5.

Properties:

KY : double
elastic ground coefficient KY in the UY direction of the global coordinate system

Available since version 2.5.

KZ : double
eleastic ground coefficient KZ in the UZ direction of the global coordinate system
Available since version 2.5.

HX : double
eleastic ground coefficient HX in the RX direction of the global coordinate system
Available since version 2.5.

Releases

Available since version 2.5.

IRobotBarEndReleaseValue

Set of value that can be assumed by the parameters of a release at bar end.

Properties:

I_BERV_NONE : = 0
none

I_BERV_STD : = 1
I_BERV_PLUS : = 2
I_BERV_MINUS : = 3
I_BERV_FIXED : = 1
Available since version 1.7.

I_BERV_ELASTIC : = 4
Available since version 1.7.

I_BERV_ELASTIC_PLUS : = 5
Available since version 1.7.

I_BERV_ELASTIC_MINUS : = 6
Available since version 1.7.

IRobotBarReleaseData

The set of data describing a release defined for a bar. The definition of the release consists of two sets of data - each one for each bar end. The data describing the entire release are stored as a data component of the labels of the following type: ILT_BAR_RELEASE.

Properties:

StartNode : IRobotBarEndReleaseData
data describing a release at bar beginning

EndNode : IRobotBarEndReleaseData
data describing a release at bar end

IRobotBarEndReleaseData

Set of parameters defining a release at one bar end. The allowable values are determined by the RobotBarEndReleaseValue type.

Properties:

UX : IRobotBarEndReleaseValue
UY : IRobotBarEndReleaseValue
UZ : IRobotBarEndReleaseValue
RX : IRobotBarEndReleaseValue
RY : IRobotBarEndReleaseValue
RZ : IRobotBarEndReleaseValue

KX : double
Available since version 1.7.

KY : double
Available since version 1.7.

KZ : double
Available since version 1.7.

HX : double
Available since version 1.7.

HY : double
Available since version 1.7.

HZ : double
Available since version 1.7.

AX : double
damping coefficient (x axis direction)
Available since version 2.5.

AY : double
damping coefficient (y axis direction)

Available since version 2.5.

AZ : double
damping coefficient (z axis direction)

Available since version 2.5.

BX : double
Bx damping coefficient
Available since version 2.5.

BY : double
By damping coefficient
Available since version 2.5.

BZ : double
Bz damping coefficient
Available since version 2.5.

NonlinearModel : IRobotNonlinearLinkMngr
non-linearity support
Available since version 3.

Geometrical imperfections

Available since version 3.

IRobotBarGeoImperfectionsData

Data set describing bar geometrical imperfections.
Available since version 3.

Operations:

SetAutomatic (_axis : IRobotBarGeoImperfectionAxis, _automatic : bool = True, _coeff : double = 1.0)
Function allows determining - for a selected axis - if initial deflection is to be calculated automatically according to EC3, or not.
Available since version 3.

IsAutomatic (_axis : IRobotBarGeoImperfectionAxis) : bool
Function returns a non-zero value (True), if for a selected axis the initial deflection is calculated automatically according to EC3.
Available since version 3.

GetBucklingCoeff (_axis : IRobotBarGeoImperfectionAxis) : double
Function returns value of the buckling length coefficient for the selected direction. This value is relevant only if deflection is calulated automatically according to EC3.
Available since version 3.

SetUser (_axis : IRobotBarGeoImperfectionAxis, _val : double)
Function sets the user-defined value of initial deflection for the indicated axis.
Available since version 3.

GetUser (_axis : IRobotBarGeoImperfectionAxis) : double
Function returns the user-defined value of initial deflection for the indicated axis.

Available since version 3.

SetMinus (_axis : IRobotBarGeoImperfectionAxis, _minus : bool = True)
Function enables change of the deflection direction for the indicated axis.
Available since version 3.

IsMinus (_axis : IRobotBarGeoImperfectionAxis) : bool
Function returns a non-zero value (True), if for the selected axis the negative deflection direction has been set.

Available since version 3.

IRobotBarGeoImperfectionsAxis

Available since version 3.

Properties:

I_BGIA_Z : = 1
Local Z axis
Available since version 3.

I_BGIA_Y : = 0
Local Y axis
Available since version 3.

Non-linear hinges

Available since version 3.

IRobotNonlinearHingeModelServer

Server of non-linear hinge models.
Available since version 3.

Properties:

Count : long
Number of available models
Available since version 3.

Operations:

Get (_idx : long) : IRobotNonlinearHingeModel
Function takes a non-linear hinge model of the indicated index.
Available since version 3.

Create (_name : string) : IRobotNonlinearHingeModel
Function creates a new non-linear model of a hinge.
Available since version 3.

Find (_name : string) : long
Function returns the model index based on the specified name.
Available since version 3.

Delete (_idx : long)
Function deletes model of the indicated index.
Available since version 3.

IRobotNonlinearHingeModel

Definition of non-linear hinge model

Available since version 3.

Properties:

Name : string
Model name
Available since version 3.

ModelType : IRobotNonlinearHingeModelType
Type of a non-linear hinge model
Available since version 3.

Symetry : bool
curve symmetry
Available since version 3.

LimitCoordX : bool
flag enforcing automatic estimation of a limit value for the quantity represented by abscissa; Example: for a model of the Force -Displacement type, it involves estimation of the force limit value.
Available since version 3.

LimitCoordY : bool
flag enforcing automatic estimation of a limit value for the quantity represented by ordinate; Example: for a model of the Force -Displacement type, it involves estimation of the displacement limit value.

Available since version 3.

LimitCoordXValue : double
coefficient used for defining a limit value for the quantity represented by abscissa
Available since version 3.

LimitCoordYValue : double
coefficient used for defining a limit value for the quantity represented by ordinate
Available since version 3.

UnloadingMethod : IRobotNonlinearHingeModelUnloadingType
Unloading method
Available since version 3.

MixedUnloadingValue : double
Value of the unloading factor for MIXED method
Available since version 3.

Operations:

GetPoints (_axis : IRobotNonlinearLinkSemiAxisType) : IRobotNonlinearHingeModelPoints
Function takes the interface for definition of curve points.
Available since version 3.

SetPoints (_axis : IRobotNonlinearLinkSemiAxisType, _points : IRobotNonlinearHingeModelPoints)
Function updates a curve definition based on the specified interface.
Available since version 3.

GetAxisParams (_axis : IRobotNonlinearLinkSemiAxisType) : IRobotNonlinearHingeModeAxisParams
Function takes the interface of the parameters characteristic of the indicated semi-axis.
Available since version 3.

SetAxisParams (_axis : IRobotNonlinearLinkSemiAxisType, _params : IRobotNonlinearHingeModelAxisParams)
Function updates parameters for the indicated semi-axis.
Available since version 3.

IRobotNonlinearHingeModelAxisParams

Interface of parameters whose values depend on the semi-axis type.
Available since version 3.

Properties:

ImmediateOccupancy : double
immediate occupancy
Available since version 3.

LifeSafety : double
life safety
Available since version 3.

StructuralStability : double
structural stability
Available since version 3.

IRobotNonlinearHingeModelType

Types of non-linear hinge models
Available since version 3.

Properties:

I_NHMT_FORCE_DISPLACEMENT : = 1
force - displacement
Available since version 3.

I_NHMT_MOMENT_ROTATION : = 2
moment - rotation

Available since version 3.

I_NHMT_STRESS_PAIN : = 3
stress - strain
Available since version 3.

IRobotNonlinearHingeModelPoints

Interface for definition of curve points.
Available since version 3.

Properties:

Count : long = 4
number of points defining a curve
Available since version 3.

Operations:

Get (_idx : long) : IRobotGeoPoint2D
Function takes the interface for definition of a point of the indicated index.
Available since version 3.

Set (_idx : long, _point : IRobotGeoPoint2D)
Function updates point definition based on the indicated interface.
Available since version 3.

IRobotNonlinearHingeModelUnloadingType

Types of unloading methods
Available since version 3.

Properties:

I_NHMUT_ELASTIC : = 1

Available since version 3.

I_NHMUT_PLASTIC : = 2

Available since version 3.

I_NHMUT_DAMAGE : = 3

Available since version 3.

I_NHMUT_MIXED : = 4

Available since version 3.

IRobotNonlinearHingeData

Interface of non-linear hinge definition.
Available since version 3.

Properties:

NormalStress : bool
flag enforcing analysis of normal stresses in a bar section with the interactions between individual forces and moments considered
Available since version 3.

Operations:

IsActive (_cmpnt : IRobotNonlinearHingeComponentType) : bool
Function checks if the indicated component participates in a hinge definition.
Available since version 3.

SetModel (_cmpnt : IRobotNonlinearHingeComponentType, _model : string)
Function ascribes a model of the indicated name to the component.
Available since version 3.

GetModel (_cmpnt : IRobotNonlinearHingeComponentType) : string
Function takes a model name for the indicated component.
Available since version 3.

Remove (_cmpnt : IRobotNonlinearHingeComponentType)
Function excludes the indicated component from the hinge definition.
Available since version 3.

IRobotNonlinearHingeComponentType

Types of components characterizing the non-linear hinge.
Available since version 3.

Properties:

I_NHCT_FX : = 1

Available since version 3.

I_NHCT_FY : = 2

Available since version 3.

I_NHCT_FZ : = 3

Available since version 3.

I_NHCT_MX : = 4

Available since version 3.

I_NHCT_MY : = 5

Available since version 3.

I_NHCT_MZ : = 6

Available since version 3.

I_NHCT_SX : = 7

Available since version 3.

IRobotNonlinearHingeDef

Interface describing a hinge defined in the structure.
Available since version 3.

Properties:

LabelName : string
name of the label storing hinge data
Available since version 3.

Offset : long
coordinate determining the point on a bar where hinge is defined
Available since version 3.

Relative : bool
flag defining the manner of interpreting the Offset value: true value indicates that the offset coordinate is a relative value of the position on a bar, contained in the interval <0, 1>; false value indicates that the offset is treated as an absolute distance from the bar beginning - interval: <0, bar_length>
Available since version 3.

Bar : long
index of the bar on which a hinge is defined
Available since version 3.

IRobotNonlinearHingeServer

Server providing support for hinge definitions in a structure.
Available since version 3.

Operations:

Count (_bar_number : long = 0) : long
Function returns the number of hinges defined on the indicated bar (local mode) or defined in a whole structure (global mode).
Available since version 3.

Get (_hinge_idx : long, _bar_number : long = 0) : IRobotNonlinearHingeDef
Function returns the interface of hinge definition in a structure. It enables work in two modes: in the local mode, by providing the index of a hinge on a bar and bar number and in the global mode, by providing the global index of a hinge in a structure.
Available since version 3.

Set (_bar_number : long, _label : string, _offset : double, _relative : bool) : long
Function defines a hinge on the indicated bar with the indicated offset and returns the global (with respect to a whole structure) index of hinge definition.
Available since version 3.

Remove (_hinge_idx : long, _bar_number : long = 0)
Function deletes a definition of a non-linear hinge. It enables work in two modes: in the local mode, by providing the local index of hinge definition and bar number and in the global mode, by providing the global index of hinge definition.
Available since version 3.

Cladding

Available since version 3.5.

IRobotCladdingData

Cladding definition interface.
Available since version 3.5.

Properties:

Type : IRobotCladdingType
cladding type
Available since version 3.5.

TrussBars : bool
truss bars are ignored during load distribution
Available since version 3.5.

Color : long
cladding color
Available since version 3.5.

Method : IRobotCladdingMethod
Method of load distribution
Available since version 8.1.

IRobotCladdingType

Cladding type.
Available since version 3.5.

Properties:

I_CT_X : = 0
anisotropic parallel to local x axis
Available since version 3.5.

I_CT_Y : = 1
anisotropic perpendicular to local x axis
Available since version 3.5.

I_CT_XY : = 2
isotropic
Available since version 3.5.

IRobotCladdingMethod

Method of load ditribution
Available since version 8.1.

Properties:

I_CM_TRAPEZOIDAL : = 0
Trapezoidal and triangular method
Available since version 8.1.

I_CM_GRILLAGE : = 1
Grillage analysis method
Available since version 8.1.

I_CM_SLAB_ANALYSIS : = 2
Slab analysis method
Available since version 8.1.

Load cases

Robot Object Model makes it possible to perform operations on load cases available in the Robot program. There are two basic types of load cases: a simple case and combinations. A simple case definition consists of a list of load records, while a combination contains a list of simple cases with the relevant coefficients.

IRobotCase

The part of the load case interface that is shared by simple cases and combinations has been defined using RobotCase.

Properties:

Number : long
user-defined number linked with the load case

Type : IRobotCaseType
load case type - allows one to differentiate between a simple case and combination at the level of the shared part of the interface

Name : string
user-defined case name

Nature : IRobotCaseNature
case nature

AnalizeType : IRobotCaseAnalizeType
analysis type

IRobotCaseNature

The set of case natures accepted by Robot is determined.

Properties:

I_CN_PERMANENT : = 0
dead

I_CN_TEMPERATURE : = 4
temperature

I_CN_EXPLOATATION : = 1
live

I_CN_WIND : = 2
wind

I_CN_SNOW : = 3
snow

I_CN_SEISMIC : = 6
seismic

I_CN_ACCIDENTAL : = 5
Available since version 1.7.

IRobotCaseAnalizeType

Robot Object Model defines a set of analysis types for load cases accepted by Robot.

Properties:

I_CAT_COMB_NONLINEAR_INCREMENTAL : = -5
Available since version 2.0.

I_CAT_COMB_BUCKLING : = -4
I_CAT_COMB_NONLINEAR_BUCKLING : = -3
I_CAT_COMB_INCREMENTAL : = -2
Available since version 2.0.

I_CAT_COMB_NONLINEAR : = -1
I_CAT_COMB : = 0
I_CAT_STATIC_LINEAR : = 1
I_CAT_STATIC_NONLINEAR : = 2
I_CAT_STATIC_INCREMENTAL : = 3
Available since version 2.0.

I_CAT_STATIC_FLAMBEMENT : = 4
Available since version 2.0.

I_CAT_STATIC_LINEAR_AUXILIARY : = 5
I_CAT_STATIC_NONLINEAR_INCREMENTAL : = 6
Available since version 2.0.

I_CAT_STATIC_NONLINEAR_FLAMBEMENT : = 7
Available since version 2.0.

I_CAT_STATIC_NONLINEAR_MODAL : = 8
modal analysis with nonlinear static state

I_CAT_DYNAMIC_MODAL : = 11
I_CAT_DYNAMIC_SPECTRAL : = 12
I_CAT_DYNAMIC_SEISMIC : = 13
I_CAT_DYNAMIC_HARMONIC : = 14
I_CAT_TEMPORAR : = 20
Time history analysis

I_CAT_MOBILE_MAIN : = 30
I_CAT_MOBILE_DERIVED : = 31
mobil analysis - derived case

I_CAT_NULL : = 99
null load case

I_CAT_COMB_CODE : = -12
code combination
Available since version 1.7.

I_CAT_STATIC_BUCKLING : = 4
Available since version 2.0.

I_CAT_STATIC_NONLINEAR_BUCKLING : = 7
Available since version 2.0.

I_CAT_DYNAMIC_NONLINEAR_MODAL_WITH_STATIC_FORCE : = 8
Available since version 2.0.

I_CAT_TIME_HISTORY : = 20
Time history analysis
Available since version 3.

I_CAT_PUSH_OVER : = 25

Available since version 3.

IRobotCaseType

Among load cases, one distinguishes three basic types: a simple case, a combination and a code combination.

Properties:

I_CT_SIMPLE : = 0
simple case

I_CT_COMBINATION : = 1
combination

I_CT_CODE_COMBINATION : = 2
code combination (defined according to a definition of the selected code)
Available since version 1.7.

I_CT_MOBILE : = 3
moving load case
Available since version 2.5.

IRobotCaseServer

Server of load cases

Properties:

FreeNumber : long
first available (free) user's number for a load case
Available since version 1.7.

SnowWindEngine : IRobotSnowWindEngine
generator of snow/wind loads
Available since version 1.7.

CodeCmbEngine : IRobotCodeCombinationEngine
generator of code-dependent loads
Available since version 1.7.

TimeHistoryFunctions : IRobotTimeHistoryFunctionList
List of available time functions for time history analysis
Available since version 3.

QCmbTau : double

Operations:

CreateSimple (number : long, name : string, nature : IRobotCaseNature, analize_type : IRobotCaseAnalizeType) : IRobotSimpleCase
The function creates and returns a simple load case with the defined parameters.

CreateCombination (num : long, name : string, type : IRobotCombinationType, nature : IRobotCaseNature, analize_type : IRobotCaseAnalizeType) : IRobotCaseCombination
The function creates and returns combinations with the defined parameters.

Exist (number : long) : bool
The function checks if there exists a load case with the user-defined number.

Get (number : long) : IRobotCase
The function returns a load case with the user-defined number. The type of the returned object depends on the real case type with the indicated number. If the indicated number defines a combination, the function returns an object of the RobotCaseCombination object type; otherwise, it will return RobotSimpleCase object type.
If the case with the indicated number does not exist, running this function leads to critical error.

GetMany (case_selection : IRobotSelection) : IRobotCaseCollection
The function returns a collection of load cases that meet the criterion of selection introduced as argument.

GetAll () : IRobotCaseCollection

The function returns a collection of all load cases defined for a given structure.

Delete (number : long)
The function deletes from the structure model the load case with the indicated number

DeleteMany (case_selection : IRobotSelection)
The function deletes all the load cases that meet the selection criteria introduced as argument.

FindWithId (unique_id : long) : long
Function returns a load case number with the specified unique identifier. If the load case is not found, then the function returns zero value (0).
Available since version 1.7.

CreateMobile (_number : long, _name : string, _nature : IRobotCaseNature) : IRobotMobileCase
Function creates moving load case.
Available since version 3.

IsAuxiliary (_case_num : long) : bool

Available since version 8.1.

SetAuxiliary (_case_num : long, _is_aux : bool)

Available since version 8.1.

IRobotSimpleCase

Derived from IRobotCase
A simple load case is defined using the list of load records.

Properties:

Records : IRobotLoadRecordMngr
list of load records associated with the case

UniqueId : long
unique load case identifier
Available since version 1.7.

ModesCount : long
number of modes
Available since version 1.7.

MainMode : long
number of the main mode (0 denotes lack of the main mode)

Available since version 1.7.

NatureName : string
Name of load case nature
Available since version 3.

TimeStepCount : long
number of defined steps (it concerns only cases of time history analysis)
Available since version 3.

Label : string

Available since version 8.

Operations:

GetAnalysisParams () : IDispatch
Function returns the object describing analysis parameters defined for the load case. The type of the returned object depends on the analysis type that has been assigned to this load case (e.g. for modal cases the object of the RobotModalAnalysisParams type is returned).
Available since version 2.5.

SetAnalysisParams (_params : IDispatch) : bool
Function sets analysis parameters for the load case. If the specified parameters cannot be set for this case (e.g. due to incompatibility between the type of case analysis and type of the object defining parameters), then zero value (False) is returned.
Available since version 2.5.

GetSeismicCode () : string
Function returns the name of a seismic code for which a given load case has been defined. For cases whose analysis type is different from seismic analysis, the function returns an empty character string.
Available since version 2.5.

SetNatureExt (_rgl_action : long)
Function assigns - to the case - the nature consistent with the indicated number of "action" described in the current regulations.
Available since version 3.

IsAuxiliary : bool

Available since version 8.1.

IRobotCaseFactorMngr

Interface granting access to the list of pairs (load case, coefficient). All pairs from the list are numbered (indexed) with numbers from 1 to the number of items (Count).

Properties:

Count : long
number of managed items

Operations:

New (case_number : long, factor : double) : long
The function creates a new pair (load case, coefficient) and adds to the list of managed items. It returns the index of the created pair.

Get (index : long) : IRobotCaseFactor
The function returns the item with the indicated index.

Delete (index : long)
The function deletes items with the defined index.

IRobotCaseCombination

Derived from IRobotCase
A combination is defined by means of simple load cases with appropriate factors.

Properties:

CaseFactors : IRobotCaseFactorMngr
list of pairs (simple case, factor) defining a combination

CombinationType : IRobotCombinationType
combination type

SeismicType : IRobotModeCombinationType
seismic combination type

Quadratic : bool
indicator informing whether we are dealing with a quadratic combination

UniqueId : long
unique identifier of load case combinations
Available since version 1.7.

Signed : bool

a flag indicating whether the combination is signed or not
Available since version 1.7.

NatureName : string
name of nature
Available since version 3.

Label : string

Available since version 8.

Operations:

GetAnalysisParams () : IDispatch
Function returns the object describing analysis parameters defined for the load case. The type of the returned object depends on the analysis type that has been assigned to this load case (e.g. for modal cases the object of the RobotModalAnalysisParams type is returned).

Available since version 2.5.

SetAnalysisParams (_params : IDispatch) : bool
Function sets analysis parameters for the load case. If the parameters specified cannot be set for this case (e.g. due to incompatibility between the case analysis type and the type of the object defining parameters), then zero value (False) is returned.

Available since version 2.5.

SetNatureExt (_rgl_action : long)
Function assigns - to the combination - the nature consistent with the indicated number of "action" described in the current regulations.

Available since version 3.

IsAuxiliary : bool

Available since version 8.1.

IRobotCombinationType

Type of load case combinations available in Robot.

Properties:

I_CBT_EFF : = 0
ultimate limit state

I_CBT_DEP : = 1
serviceability limit state

I_CBT_ACC : = 2
accidental state

I_CBT_SPC : = 3
special state

I_CBT_ULS : = 0
ultimate limit state
Available since version 2.0.

I_CBT_SLS : = 1
serviceability limit state
Available since version 2.0.

I_CBT_ALS : = 2
accidental limit state
Available since version 2.0.

IRobotCaseFactor

Interface describing a pair (load case, coefficient) used in the definition of load case combination.

Properties:

CaseNumber : long
load case number

Factor : double
(multiplication) factor

IRobotCaseCollection

Derived from IRobotCollection
Collection of load cases

IRobotLimitState

Limit state type.

Properties:

I_LS_ULS : = 0
ultimate limit state

I_LS_SLS : = 1
serviceability limit state

I_LS_ALS : = 2
accidental state

I_LS_SPC : = 3

IRobotCaseAnalysisModesFilter

Definition of modes taken into account during structure dynamic analysis.
Available since version 2.5.

Properties:

Type : IRobotCaseAnalysisModesFilterType
filtering type
Available since version 2.5.

MassPercentage : double

Available since version 2.5.

Modes : BSTR
list of modes taken into account during structure dynamic analysis.
Available since version 2.5.

IRobotCaseAnalysisModesFilterType

Available methods of mode filtering
Available since version 2.5.

Properties:

I_CAMFT_NON_ACTIVE : = 0
inactive filtering
Available since version 2.5.

I_CAMFT_MASS_PERCENTAGE : = 1
during structure analysis only these modes will be taken into account whose value of the mass participation factor is greater than the specified one
Available since version 2.5.

I_CAMFT_LIST_OF_MODES : = 2
during structure analysis only the modes specified by the user will be taken into account
Available since version 2.5.

IRobotDynamicAnalysisExcitationDirection

Available since version 5.5.

Properties:

X : double
Y : double
Z : double
UseNormalized : bool
NormalizedX : double
NormalizedY : double
NormalizedZ : double
ResolutionActive : bool
QuadraticActive : bool
Rx : double
Ry : double
Rz : double
QuadraticSigned : bool
Group1 : bool
Group2 : bool
Group3 : bool
Mi : double
Lambda : double
CombType : IRobotModeCombinationType

Load records

A load record is used to model loads applied to a structure

IRobotLoadRecordType

Load record types available in the Robot Object Model

Properties:

I_LRT_NODE_FORCE : = 0
I_LRT_NODE_DISPLACEMENT : = 1
I_LRT_BAR_DILATATION : = 2
I_LRT_BAR_FORCE_CONCENTRATED : = 3
I_LRT_BAR_MOMENT_DISTRIBUTED : = 4
I_LRT_BAR_UNIFORM : = 5
I_LRT_BAR_TRAPEZOIDALE : = 6
I_LRT_BAR_DEAD : = 7
I_LRT_BAR_THERMAL : = 8
I_LRT_LINEAR : = 21
linear load

I_LRT_LINEAR_3D : = 19
linear load defined in 3D space

I_LRT_NODE_AUXILIARY : = 20
I_LRT_POINT_AUXILIARY : = 23
I_LRT_IN_3_POINTS : = 22
planar load defined at three points

I_LRT_PRESSURE : = 24
hydrostatic pressure

I_LRT_THERMAL_IN_3_POINTS : = 25
planar thermal load defined at three points

I_LRT_IN_CONTOUR : = 28
planar load defined on the contour (uniform or given at three points)

I_LRT_NODE_FORCE_MASS : = 30
I_LRT_BAR_FORCE_CONCENTRATED_MASS : = 33
I_LRT_BAR_UNIFORM_MASS : = 35
I_LRT_BAR_TRAPEZOIDALE_MASS : = 36
I_LRT_MASS_ACTIVATION : = 39
I_LRT_SPECTRUM_VALUE : = 40
I_LRT_UNIFORM : = 26
uniform surface load

I_LRT_THERMAL : = 8
I_LRT_NODE_FORCE_IN_POINT : = 23
concentrated force in the point
Available since version 1.7.

I_LRT_LINEAR_ON_EDGES : = 69
linear load on edges
Available since version 1.7.

I_LRT_DEAD : = 7
self-weight
Available since version 2.0.

I_LRT_SURFACE_ON_OBJECT : = 70
Surface load on object
Available since version 3.

I_LRT_MOBILE_POINT_FORCE : = 53
Moving load - point force
Available since version 3.

I_LRT_MOBILE_DISTRIBUTED : = 55
a moving distributed load:
- linear, if value Dx or Dy equals zero
- planar, if Dx and Dy assume non-zero values

Available since version 3.

IRobotLoadRecord

The load record is used to model a load applied to a structure. RobotLoadRecord is described by an interface shared by all load record types available in Robot.

Properties:

Type : IRobotLoadRecordType
load record type.

ObjectType : IRobotObjectType
type of the object where a load is applied

Objects : IRobotSelection
selection of objects - structure components where a load is applied. The object type described by the selection corresponds to the type of objects to which the load may be applied.

Description : string
load record description

Operations:

GetValue (value_id : short) : double
The function returns the value of the indicated quantity describing a load
The definition of a load record consists of a set of numbers - values describing the modeled load. The type and significance of particular values depend on the load record type and they are referred to by an appropriate identifier.

SetValue (value_id : short, value : double)
The function sets the value of the indicated quantity describing a load. The definition of a load record consists of a set of numbers - values describing the modeled load. The type and significance of particular values depend on the load record type and they are referred to by an appropriate identifier.

IRobotLoadRecordMngr

Load record manager manages logical lists of inter-linked records (e.g. a list of records linked with the same load case). The records are numbered from 1 to the number of records in the list

Properties:

Count : long
number of currently managed load records

Operations:

New (record_type : IRobotLoadRecordType) : long
The function creates a new load record of the indicated type. The newly-created record is added at the end of the managed list. The function gives the index of the crated record back.

Get (record_index : long) : IRobotLoadRecord
The function returns a record with the indicated index on the list of the managed load records.

Delete (record_index : long)
The function deletes the record with the indicated index. The list of all records is automatically re-numbered after this operation.

IRobotLoadRecordLinear3D

Derived from IRobotLoadRecord
Load record ot the I_LRT_LINEAR_3D type.

Properties:

UniqueId : long
unique identifier of a load record
Available since version 1.7.

Operations:

SetPoint (which_point : int, x : double, y : double, z : double)
The function sets the coordinates of the indicated point of load application.

GetPoint (which_point : int, x : double*, y : double*, z : double*)
The function returns the coordinates of the indicated point of load application.

IRobotLoadRecordIn3Points

Derived from IRobotLoadRecord
Load record for a load defined by three points

Properties:

UniqueId : long
unique identifier of a load record
Available since version 1.7.

Operations:

SetPoint (which_point : int, x : double, y : double, z : double)
The function sets the coordinates of the indicated point of load application.

GetPoint (which_point : int, x : double*, y : double*, z : double*) : bool
The function returns coordinates of the indicated point of load application.

SetGeoLimits (layer_def : IRobotGeoLayer) : bool
Function sets geometrical limits for loads.
Available since version 2.0.

GetGeoLimits () : IRobotGeoLayer
Function returns geometrical limits applied to loads.
Available since version 2.0.

IRobotLoadRecordThermalIn3Points

Derived from IRobotLoadRecord
Thermal load in three points.

Properties:

UniqueId : long
unique identifier of a load record
Available since version 1.7.

Operations:

SetPoint (which_point : int, x : double, y : double, z : double)
The function sets the coordinates of the indicated point of load application.

GetPoint (which_point : int, x : double*, y : double*, z : double*) : bool
The function sets the coordinates of the indicated point of load application.

SetGeoLimits (layer_def : IRobotGeoLayer) : bool
Function sets geometrical limits for loads.
Available since version 2.0.

GetGeoLimits () : IRobotGeoLayer
Function returns an object describing geometrical limitations of the load. If the load is not limited, then an empty reference to the object (NULL) will be returned.
Available since version 2.0.

IRobotLoadRecordInContour

Derived from IRobotLoadRecord

Planar load on the contour. For a uniform load only the values for the first point of the load should be specified (I_ICRV_PX1, I_ICRV_PY1 and I_ICRV_PZ1) without providing the point coordinates. For a variable load all the load values and coordinates of the appropriate points should be specified applying the SetPoint() function.

Properties:

UniqueId : long

unique identifier of a load record
Available since version 1.7.

Operations:

SetVector (vx : double, vy : double, vz : double)

Function sets vector of load projection.

GetVector (vx : double*, vy : double*, vz : double*)

Function takes coordinates of a load projection vector.

SetPoint (which_point : int, x : double, y : double, z : double)

The function sets the coordinates of the indicated point of loading.Only for variable loads at three points.

GetPoint (which_point : int, x : double*, y : double*, z : double*) : bool

The function gets the coordinates of a point. If the indicated point has not been defined, the function returns False (zero value).

SetContourPoint (which_point : int, x : double, y : double, z : double)

The function sets the coordinates of the indicated point of a contour.

GetContourPoint (which_point : int, x : double*, y : double*, z : double*)

The function gets the coordinates of the indicated point of a contour.

SetGeoLimits (layer_def : IRobotGeoLayer) : bool

Function defines geometrical limitations for a load. If the limitations are defined successfully the function returns a value different from zero (True).
Available since version 2.0.

GetGeoLimits () : IRobotGeoLayer

Function returns an object describing geometrical limitations for a load. If they have not been defined, then an empty reference to the object will be returned (NULL).

Available since version 2.0.

IRobotLoadRecordLinear

Derived from IRobotLoadRecord

Properties:

UniqueId : long
unique identifier of a load record
Available since version 1.7.

Operations:

SetPoint (which_point : int, x : double, y : double, z : double)
The function sets the coordinate of the indicated point.

GetPoint (which_point : int, x : double*, y : double*, z : double*)
The function gets the coordinate of the indicated point.

IRobotLoadRecord2

Derived from IRobotLoadRecord
Extended interface of the record that describes a load.

Properties:

UniqueId : long
unique identifier of a load record

Operations:

SetGeoLimits (layer_def : IRobotGeoLayer) : bool
Function defines geometrical limitations for a load. It may happen to be the case that a definition method or load type does not allow defining geometrical limitations for it. The function returns the information if application of limitations has been successful.
In order to delete geometrical limitations, an empty reference to the object should be provided as a parameter.
Available since version 2.0.

GetGeoLimits () : IRobotGeoLayer
Function returns an object describing a geometrical limitation defined for this load. If the limitation has not been defined, the function returns an empty reference.
Available since version 2.0.

Values describing load records

Each load record is defined by means of several numerical values. The significance of particular values depends on the load record type. A set if identifiers is defined to refer to particular values in the relevant load records. In order to define correctly a load record, the appropriate set of values (defined for this record type) should be recognized first. The value identifiers are used as parameters of the SetValue() and GetValue() functions for the load record.

IRobotNodeForceRecordValues

Identifiers of the values describing load records of the following types: I_LRT_NODE_FORCE and I_LRT_NODE_FORCE_MASS.

Properties:

I_NFRV_FX : = 0
I_NFRV_FY : = 1
I_NFRV_FZ : = 2
I_NFRV_CX : = 3
I_NFRV_CY : = 4
I_NFRV_CZ : = 5
I_NFRV_ALPHA : = 8
I_NFRV_BETA : = 9
I_NFRV_GAMMA : = 10

IRobotNodeDisplacementRecordValues

The identifier of values for the records of the I_LRT_NODE_DISPLACEMENT type

Properties:

I_NDRV_UX : = 0
I_NDRV_UY : = 1
I_NDRV_UZ : = 2
I_NDRV_RX : = 3
I_NDRV_RY : = 4
I_NDRV_RZ : = 5
I_NDRV_ALPHA : = 8
I_NDRV_BETA : = 9
I_NDRV_GAMMA : = 10

IRobotNodeAuxiliaryRecordValues

Identifiers of values describing load records of the I_LRT_NODE_AUXILIARY type.

Properties:

I_NARV_FX : = 0
I_NARV_FY : = 1
I_NARV_FZ : = 2

I_NARV_CX : = 3
I_NARV_CY : = 4
I_NARV_CZ : = 5
I_NARV_NR : = 6
I_NARV_ALPHA : = 8
I_NARV_BETA : = 9
I_NARV_GAMMA : = 10

IRobotBarDilatationRecordValues

Identifiers of the values describing load records of the following type:

Properties:

I_BDRV_DIL : = 0
I_BDRV_REL : = 13

IRobotBarForceConcentrateRecordValues

Identifier of the values describing load records of the following types: I_LRT_BAR_FORCE_CONCENTRATE, I_LRT_BAR_FORCE_CONCENTRATE_MASS

Properties:

I_BFCRV_FX : = 0
I_BFCRV_FY : = 1
I_BFCRV_FZ : = 2
I_BFCRV_CX : = 3
I_BFCRV_CY : = 4
I_BFCRV_CZ : = 5
I_BFCRV_X : = 6
I_BFCRV_ALPHA : = 8
I_BFCRV_BETA : = 9
I_BFCRV_GAMMA : = 10
I_BFCRV_LOC : = 11
I_BFCRV_REL : = 13
I_BFCRV_OFFSET_Y : = 21
Offset value of the force application point (Y local direction)
Available since version 3.

I_BFCRV_OFFSET_Z : = 22
Offset value of the force application point (Z local direction)
Available since version 3.

IRobotBarMomentDistributedRecordValues

Identifiers of the values describing load records of the following type: I_LRT_BAR_MOMENT_DISTRIBUTED

Properties:

I_BMDRV_MX : = 0
I_BMDRV_MY : = 1
I_BMDRV_MZ : = 2
I_BMDRV_ALPHA : = 8
I_BMDRV_BETA : = 9
I_BMDRV_GAMMA : = 10

IRobotBarUniformRecordValues

Identifiers of the values describing load records of the following type: I_LRT_BAR_UNIFORM and I_LRT_BAR_UNIFORM_MASS

Properties:

I_BURV_PX : = 0
I_BURV_PY : = 1
I_BURV_PZ : = 2
I_BURV_ALPHA : = 8
I_BURV_BETA : = 9
I_BURV_GAMMA : = 10
I_BURV_LOCAL : = 11
I_BURV_PROJECTION : = 12
I_BURV_RELATIVE : = 13
Available since version 2.0.

I_BURV_OFFSET_Y : = 21
Offset value of the uniform load application point (Y local direction)
Available since version 3.

I_BURV_OFFSET_Z : = 22
Offset value of the uniform load application point (Z local direction)
Available since version 3.

IRobotBarTrapezoidaleRecordValues

Identifiers of the values describing load records of the following type: I_LRT_BAR_TRAPEZOIDALE and I_LRT_BAR_TRAPEZOIDALE_MASS.

Properties:

I_BTRV_PX2 : = 0
I_BTRV_PY2 : = 1
I_BTRV_PZ2 : = 2
I_BTRV_PX1 : = 3

I_BTRV_PY1 : = 4
I_BTRV_PZ1 : = 5
I_BTRV_X2 : = 6
I_BTRV_X1 : = 7
I_BTRV_ALPHA : = 8
I_BTRV_BETA : = 9
I_BTRV_GAMMA : = 10
I_BTRV_LOCAL : = 11
I_BTRV_PROJECTION : = 12
I_BTRV_RELATIVE : = 13

IRobotBarThermalRecordValues

Identifiers of the values describing load records of the following type: I_LRT_BAR_THERMAL

Properties:

I_BTRV_TX : = 0
I_BTRV_TY : = 1
I_BTRV_TZ : = 2

IRobotBarDeadRecordValues

Identifiers of the values describing load records of the following type: I_LRT_BAR_DEAD.

Properties:

I_BDRV_X : = 0
I_BDRV_Y : = 1
I_BDRV_Z : = 2
I_BDRV_ENTIRE_STRUCTURE : = 15
flag indicating that load is applied to the whole structure
Available since version 3.

IRobotPointAuxiliaryRecordValues

Identifiers of the values describing load records of the following type: I_LRT_POINT_AUXILIARY.

Properties:

I_PARV_FX : = 0
I_PARV_FY : = 1
I_PARV_FZ : = 2
I_PARV_CX : = 3
I_PARV_CY : = 4
I_PARV_CZ : = 5

I_PARV_ALPHA : = 8
I_PARV_BETA : = 9
I_PARV_GAMMA : = 10
I_PARV_X : = 11
I_PARV_Y : = 12
I_PARV_Z : = 13

IRobotIn3PointsRecordValues

Identifiers of the values describing load records of the following type: I_LRT_IN_3_POINTS.

Properties:

I_3PRV_PX1 : = 0
I_3PRV_PY1 : = 1
I_3PRV_PZ1 : = 2
I_3PRV_PX2 : = 3
I_3PRV_PY2 : = 4
I_3PRV_PZ2 : = 5
I_3PRV_PX3 : = 6
I_3PRV_PY3 : = 7
I_3PRV_PZ3 : = 8
I_3PRV_PROJECTION : = 12
I_3PRV_N1 : = 9
Available since version 1.7.

I_3PRV_N2 : = 10
Available since version 1.7.

I_3PRV_N3 : = 11
Available since version 1.7.

I_3PRV_LOCAL_SYSTEM : = 13
Available since version 2.0.

IRobotPressureRecordValues

Identifiers of the values describing load records of the following type: I_LRT_PRESSURE.

Properties:

I_PRV_P : = 0
I_PRV_RO : = 1
unit weight of liquid
Available since version 2.0.

I_PRV_H : = 2

liquid level
Available since version 2.0.

I_PRV_DIRECTION : = 6
direction of gravity for liquid
(0 : -X, 1 : -Y, 2 : -Z, 3 : X, 4 : Y, 5 : Z)

IRobotLinearRecordValues

Identifiers of the values describing load records of the following type: I_LRT_LINEAR

Properties:

I_LRV_MX1 : = 0
I_LRV_MX2 : = 1
I_LRV_PZ1 : = 2
I_LRV_PZ2 : = 3
I_LRV_MY1 : = 4
I_LRV_MY2 : = 5
I_LRV_NODE_1 : = 6
I_LRV_NODE_2 : = 7
I_LRV_LOCAL : = 11

IRobotMassActivationRecordValues

Identifiers of the values describing load records of the following type: I_LRT_MASS_ACTIVATION

Properties:

I_MARV_INPUT_DIR_X : = 0
I_MARV_INPUT_DIR_Y : = 1
I_MARV_INPUT_DIR_Z : = 2
I_MARV_FACTOR : = 3
I_MARV_CASE_NUM : = 4
I_MARV_ACTIVATION_DIR : = 6
I_MARV_SIGN : = 7

IRobotLinear3DRecordValues

Identifiers of the values describing load records of the following type: I_LRT_LINEAR_3D

Properties:

I_L3DRV_PX1 : = 0

I_L3DRV_PY1 : = 1
I_L3DRV_PZ1 : = 2
I_L3DRV_MX1 : = 3
I_L3DRV_MY1 : = 4
I_L3DRV_MZ1 : = 5
I_L3DRV_PX2 : = 6
I_L3DRV_PY2 : = 7
I_L3DRV_PZ2 : = 8
I_L3DRV_MX2 : = 9
I_L3DRV_MY2 : = 10
I_L3DRV_MZ2 : = 11
I_L3DRV_LOCAL : = 13
I_L3DRV_GAMMA : = 15

IRobotThermalIn3PointsRecordValues

Identifiers of the values describing load records of the following type: I_LRT_THERMAL_IN_3_POINTS (planar thermal load 3P).

Properties:

I_3PRV_TX1 : = 0
I_3PRV_TZ1 : = 2
I_3PRV_TX2 : = 3
I_3PRV_TZ2 : = 5
I_3PRV_TX3 : = 6
I_3PRV_TZ3 : = 8

IRobotInContourRecordValues

Identifiers of the values describing load records of the following type: I_LRT_IN_CONTOUR (planar load on contour - uniform or variable)

Properties:

I_ICRV_PX1 : = 0
I_ICRV_PY1 : = 1
I_ICRV_PZ1 : = 2
I_ICRV_PX2 : = 3
I_ICRV_PY2 : = 4
I_ICRV_PZ2 : = 5
I_ICRV_PX3 : = 6
I_ICRV_PY3 : = 7
I_ICRV_PZ3 : = 8
I_ICRV_PROJECTION : = 12
flag indicating whether the load is to be projected (value different from zero) or not (zero)

I_ICRV_NPOINTS : = 13
number of points defining a contour

I_ICRV_LOCAL : = 15
variable describing coordinate system: local (the value not equal zero) or global (zero)

I_ICRV_AUTO_DETECT_OBJECTS : = -1

Available since version 8.1.

IRobotUniformRecordValues

Identifiers of the values describing load records of the following type: I_LRT_UNIFORM

Properties:

I_URV_PX : = 0
I_URV_PY : = 1
I_URV_PZ : = 2
I_URV_LOCAL_SYSTEM : = 11
parameter assumes the value 0 (zero) when the load acts in the global system and the value different from zero when the load acts in the local system

I_URV_PROJECTED : = 12
Available since version 2.0.

I_URV_RELATIVE : = 13
Available since version 2.0.

IRobotThermalRecordValues

Identifiers of the values describing load records of the following type: I_LRT_THERMAL (thermal load: uniform, linear, at three points)

Properties:

I_TRV_T_1 : = 0
Available since version 2.0.

I_TRV_GRADIENT_1 : = 2
Available since version 2.0.

I_TRV_T_2 : = 3
Available since version 2.0.

I_TRV_GRADIENT_2 : = 5
Available since version 2.0.

I_TRV_T_3 : = 6
Available since version 2.0.

I_TRV_GRADIENT_3 : = 8
Available since version 2.0.

IRobotNodeForceInPointRecordValues

Values that define a load resulting from a force acting at a point (load record type : I_LRT_NODE_FORCE_IN_POINT).

Properties:

I_NFIPRV_FX : = 0
a value of concentrated force acting along the X direction
Available since version 1.7.

I_NFIPRV_FY : = 1
a value of concentrated force acting along the Y direction
Available since version 1.7.

I_NFIPRV_FZ : = 2
a value of concentrated force acting along the Z direction
Available since version 1.7.

I_NFIPRV_MX : = 3
a value of moment (about X axis)
Available since version 1.7.

I_NFIPRV_MY : = 4
a value of moment (about Y axis)
Available since version 1.7.

I_NFIPRV_MZ : = 5
a value of moment (about Z axis)
Available since version 1.7.

I_NFIPRV_ALPHA : = 8
Available since version 1.7.

I_NFIPRV_BETA : = 9
Available since version 1.7.

I_NFIPRV_GAMMA : = 10
Available since version 1.7.

I_NFIPRV_POINT_X : = 11
X coordinate of the point of load application
Available since version 1.7.

I_NFIPRV_POINT_Y : = 12
Y coordinate of the point of load application
Available since version 1.7.

I_NFIPRV_POINT_Z : = 13
Z coordinate of the point of load application
Available since version 1.7.

IRobotLinearOnEdgesRecordValues

Parameters of linear load on edges

Properties:

I_LOERV_PX : = 0
Available since version 2.0.

I_LOERV_PY : = 1
Available since version 2.0.

I_LOERV_PZ : = 2
Available since version 2.0.

I_LOERV_MX : = 3
Available since version 2.0.

I_LOERV_MY : = 4
Available since version 2.0.

I_LOERV_MZ : = 5
Available since version 2.0.

I_LOERV_GAMMA : = 6
Available since version 2.0.

I_LOERV_LOCAL_SYSTEM : = 11
Available since version 2.0.

IRobotDeadRecordValues

Available since version 3.

Properties:

I_DRV_X : = 0

Available since version 3.

I_DRV_Y : = 1

Available since version 3.

I_DRV_Z : = 2

Available since version 3.

I_DRV_ENTIRE_STRUCTURE : = 15

Available since version 3.

IRobotSurfaceOnObjectRecordValues

Indentifiers of values for records of I_LRT_SURFACE_ON_OBJECT type.
Available since version 3.

Properties:

I_SOORV_PX : = 0

Available since version 3.

I_SOORV_PY : = 1

Available since version 3.

I_SOORV_PZ : = 2

Available since version 3.

I_SOORV_LOCAL : = 11

Available since version 3.

IRobotMobilePointForceRecordValues

Values describing a load record of the I_LRT_MOBILE_POINT_FORCE type.
Available since version 3.

Properties:

I_MPFRV_X : = 0
X coordinate of load application point
Available since version 3.

I_MPFRV_Y : = 1
Y coordinate of load application point
Available since version 3.

I_MPFRV_Z : = 2
Z coordinate of load application point
Available since version 3.

I_MPFRV_FX : = 3

Available since version 3.

I_MPFRV_FY : = 4

Available since version 3.

I_MPFRV_FZ : = 5

Available since version 3.

IRobotMobileDistributedRecordValues

Values describing a load of the I_LRT_MOBILE_DISTRIBUTED type.
Available since version 3.

Properties:

I_MDRV_X : = 0
X coordinate of load application point
Available since version 3.

I_MDRV_Y : = 1
Y coordinate of load application point
Available since version 3.

I_MDRV_Z : = 2
Z coordinate of load application point
Available since version 3.

I_MDRV_PX : = 3

Available since version 3.

I_MDRV_PY : = 4

Available since version 3.

I_MDRV_PZ : = 5

Available since version 3.

I_MDRV_DX : = 6

length of a segment/vehicle side along which the load is acting (along the vehicle axis)
Available since version 3.

I_MDRV_DY : = 7
length of a segment/vehicle side along which the load is acting (perpendicular to the vehicle axis)
Available since version 3.

Modal analysis parameters

IRobotModalAnalysisMode

Available types of modal analysis.
Available since version 2.5.

Properties:

I_MAM_MODAL : = 0
modal analysis
Available since version 2.5.

I_MAM_SEISMIC : = 1
seismic analysis
Available since version 2.5.

I_MAM_SEISMIC_PSEUDO : = 2
pseudomodal seismic analysis
Available since version 2.5.

IRobotModalAnalysisAlgorithm

Available methods of eigenproblem solution.
Available since version 2.5.

Properties:

I_MAA_BLOCK_SUBSPACE_ITERATION : = 0
block subspace iteration
Available since version 2.5.

I_MAA_SUBSPACE_ITERATION : = 1
subspace iteration
Available since version 2.5.

I_MAA_LANCZOS : = 2
Lanczos method
Available since version 2.5.

I_MAA_BASE_REDUCTION : = 3
Base reduction method
Available since version 2.5.

I_MAA_PCG_RITZ : = 4
Ritz method
Available since version 2.5.

I_MAA_PCG : = 5
preconditioned conjugate gradient method
Available since version 2.5.

IRobotModalAnalysisMassMatrixType

Available mass matrix types
Available since version 2.5.

Properties:

I_MAMMT_CONSISTENT : = 0
consistent mass matrix
Available since version 2.5.

I_MAMMT_LUMPED_WITH_ROTATIONS : = 1
lumped mass matrix with rotations
Available since version 2.5.

I_MAMMT_LUMPED : = 2
lumped mass matrix without rotations
Available since version 2.5.

IRobotModalAnalysisLimitType

Available quantities by means of which a limit value may be defined.
Available since version 2.5.

Properties:

I_MALT_PULSATION : = 0
pulsation
Available since version 2.5.

I_MALT_FREQUENCY : = 1
frequency
Available since version 2.5.

I_MALT_PERIOD : = 2
period

Available since version 2.5.

IRobotModalAnalysisParams

Parameters of modal analysis
Available since version 2.5.

Properties:

Mode : IRobotModalAnalysisMode
Analysis mode
Available since version 2.5.

ModesCount : long
number of modes
Available since version 2.5.

Tolerance : double
calculation tolerance value
Available since version 2.5.

IterationsCount : long
number of iterations
Available since version 2.5.

Damping : double
damping value
Available since version 2.5.

IncludeDampingInCalculations : bool
flag indicating if damping is to be taken into account in calculations
Available since version 2.5.

MassMatrix : IRobotModalAnalysisMassMatrixType
mass matrix
Available since version 2.5.

MassParticipation : double
definition of the value of mass participation percentage
Available since version 2.5.

DisregardDensity : bool
flag indicating if density of the structure element is to be taken into account during analysis
Available since version 2.5.

Method : IRobotModalAnalysisAlgorithm
method of eigenproblem solution
Note:

The selected algorithm of eigenproblem solution must be compatible with the active solver.

Available since version 2.5.

Shifts : IRobotModalAnalysisShifts
shift definition
Available since version 2.5.

Acceleration : double
gravity value
Available since version 2.5.

Limits : IRobotModalAnalysisLimits
limitation of the calculated mode number
Available since version 2.5.

SturmVerification : bool
switching on or off Sturm Check
Available since version 2.5.

Base : IRobotModalAnalysisBase
reduced base definition
Available since version 2.5.

MassEccentricities : IRobotMassEccentricities
definition of eccentricities
Available since version 6.21.

IRobotModalAnalysisBase

Reduced base definition.
Available since version 2.5.

Operations:

Add (_nodes : string, _ux : bool, _uy : bool, _uz : bool) : long
Function generates a new element defining the base and returns its index.
Available since version 2.5.

Delete (_idx : long) : bool
Function deletes the indicated element defining the base.
Available since version 2.5.

Get (_idx : long, _nodes : BSTR*, _ux : long*, _uy : long*, _uz : long*)
Function returns the selected element defining the base.
Available since version 2.5.

Count () : long

Function returns the number of elements defining the base. Elements defining the base are indexed from 1 to Count.
Available since version 2.5.

IRobotModalAnalysisLimits

Definition of a limit range.
Available since version 2.5.

Properties:

PulsationLimitValue : double
limit pulsation value
Available since version 2.5.

FrequencyLimitValue : double
limit frequency value
Available since version 2.5.

PeriodLimitValue : double
limit period value
Available since version 2.5.

Operations:

DefineLimits (_limit_type : IRobotModalAnalysisLimitType, _value : double)
Function defines the limit range based on the specified parameters.
Available since version 2.5.

IRobotModalAnalysisShifts

Shift definition.
Available since version 2.5.

Properties:

IterationsCount : long
number of iterations between shifts
Available since version 2.5.

Operations:

SetDefault ()
Function sets the default value for the number of iterations between shifts.
Available since version 2.5.

IRobotMassEccentricities

Definition of eccentricities.
Available since version 6.21.

Properties:

RelativeValues : bool
flag indicating if the specified values are relative or absolute
Available since version 6.21.

IsDirX : bool
flag indicating if the value for the X direction is defined
Available since version 6.21.

IsDirY : bool
flag indicating if the value for the Y direction is defined
Available since version 6.21.

ValueDirX : double
value for the X direction
Available since version 6.21.

ValueDirY : double
value for the Y direction
Available since version 6.21.

Modal analysis parameters taking into account static forces

Available since version 2.5.

IRobotModalWithStaticForcesAnalysisParams

Derived from IRobotModalAnalysisParams
Modal analysis parameters with static forces taken into account.
Available since version 2.5.

Properties:

Nonlinear : bool
non-linear analysis
Available since version 2.5.

Operations:

GetStaticStateParams () : IRobotNonlinearAnalysisParams
Function returns parameters of non-linear analysis.
Available since version 2.5.

SetStaticStateParams (_params : IRobotNonlinearAnalysisParams)
Function sets parameters of non-linear analysis.
Available since version 2.5.

CreateFromStatic (_static_case_num : long) : bool
Function initializes parameters of modal analysis with static forces taken into account based on the specified static load case.
Available since version 2.5.

Seismic analysis parameters

Available since version 2.5.

IRobotSeismicAnalysis_AFPS_90_Params

Seismic analysis parameters for the French AFPS code.
Available since version 2.5.

Properties:

Direction : IRobotGeoPoint3D
seismic excitation direction
Available since version 2.5.

Filter : IRobotCaseAnalysisModesFilter
description of modes taken into account during structure dynamic analysis
Available since version 2.5.

ZoneType : IRobotSeismicAnalysis_AFPS_90_ZoneType
seismic zone number
Available since version 2.5.

Topography : double
topography
Available since version 2.5.

BehaviorFactor : double
behavior factor
Available since version 2.5.

StructureType : IRobotSeismicAnalysis_AFPS_90_StructureType
structure type
Available since version 2.5.

Site : IRobotSeismicAnalysis_AFPS_90_SiteType
structure site type
Available since version 2.5.

SpectrumType : IRobotSeismicAnalysisSpectrumType
spectrum type
Available since version 2.5.

DirectionType : IRobotSeismicAnalysisDirectionType
direction
Available since version 2.5.

IRobotSeismicAnalysis_CHINESE_Params

Seismic analysis parameters for the Chinese code.
Available since version 2.5.

Properties:

Direction : IRobotGeoPoint3D
seismic excitation direction
Available since version 2.5.

Filter : IRobotCaseAnalysisModesFilter
description of modes taken into account during structure dynamic analysis
Available since version 2.5.

StructureType : IRobotSeismicAnalysis_CHINESE_StructureType
structure type
Available since version 2.5.

Site : IRobotSeismicAnalysis_CHINESE_SiteType
site
Available since version 2.5.

DesignStandard : IRobotSeismicAnalysis_CHINESE_DesignType
design standard
Available since version 2.5.

Intensity : IRobotSeismicAnalysis_CHINESE_IntensityType
intensity
Available since version 2.5.

EarthquakeType : IRobotSeismicAnalysis_CHINESE_EarthquakeType
earthquake distance
Available since version 2.5.

Factor : double
factor
Available since version 2.5.

SiteTg : double
site

Available since version 2.5.

IRobotSeismicAnalysis_CIRSOC_103_Params

Seismic analysis parameters for the Argentinian code.
Available since version 2.5.

Properties:

Direction : IRobotGeoPoint3D
seismic excitation direction
Available since version 2.5.

Filter : IRobotCaseAnalysisModesFilter
description of modes taken into account during structure dynamic analysis
Available since version 2.5.

ZoneType : IRobotSeismicAnalysis_CIRSOC_103_ZoneType
seismic zone number
Available since version 2.5.

DirectionType : IRobotSeismicAnalysisDirectionType
direction
Available since version 2.5.

StructureType : IRobotSeismicAnalysis_CIRSOC_103_StructureType
structure type
Available since version 2.5.

PlasticityCoeff : double
global plasticity coefficient
Available since version 2.5.

Soil : IRobotSeismicAnalysi_CIRSOC_103_SoilType

Available since version 2.5.

IRobotSeismicAnalysis_DM_16_1_96_Params

Seismic analysis parameters for the Italian DM 16.1.96 code.
Available since version 2.5.

Properties:

Direction : IRobotGeoPoint3D
seismic excitation direction
Available since version 2.5.

Filter : IRobotCaseAnalysisModesFilter

description of modes taken into account during structure dynamic analysis
Available since version 2.5.

SeismicProtectionCoeff : IRobotSeismicAnalysisProtectionCoeffType
seismic protection coefficient
Available since version 2.5.

SeismicCoeff : double
seismic coefficient
Available since version 2.5.

IRobotSeismicAnalysis_EC_8_Params

Seismic analysis parameters for Eurocode 8.
Available since version 2.5.

Properties:

Direction : IRobotGeoPoint3D
seismic excitation direction
Available since version 2.5.

Filter : IRobotCaseAnalysisModesFilter
description of modes taken into account during structure dynamic analysis
Available since version 2.5.

ZoneType : IRobotSeismicAnalysis_EC_8_ZoneType
seismic zone number
Available since version 2.5.

StructureType : IRobotSeismicAnalysis_EC_8_StructureType
structure type
Available since version 2.5.

SoilClass : IRobotSeismicAnalysis_EC_8_SoilType
soil class
Available since version 2.5.

SpectrumType : IRobotSeismicAnalysisSpectrumType
spectrum
Available since version 2.5.

DirectionType : IRobotSeismicAnalysisDirectionType
direction
Available since version 2.5.

BehaviorFactor : double
behavior factor
Available since version 2.5.

IRobotSeismicAnalysis_P_100_92_Params

Seismic analysis parameters for the Romanian P100-92 code.
Available since version 2.5.

Properties:

Direction : IRobotGeoPoint3D
seismic excitation direction
Available since version 2.5.

Filter : IRobotCaseAnalysisModesFilter
description of modes taken into account during structure dynamic analysis
Available since version 2.5.

ZoneType : IRobotSeismicAnalysis_P100_92_ZoneType
sesmic zone number
Available since version 2.5.

ImportanceClass : IRobotSeismicAnalysis_P_100_92_ImportanceClassType
importance factor
Available since version 2.5.

Tc : double

Available since version 2.5.

Psi : double

Available since version 2.5.

ExcitationDir : IRobotDynamicAnalysisExcitationDirection

IRobotSeismicAnalysis_PS_69_Params

Seismic analysis parameters for the French PS69 code.
Available since version 2.5.

Properties:

Direction : IRobotGeoPoint3D
seismic excitation direction
Available since version 2.5.

Filter : IRobotCaseAnalysisModesFilter
description of modes taken into account during structure dynamic analysis
Available since version 2.5.

Damping : IRobotSeismicAnalysis_PS_69_DampingType
damping

Available since version 2.5.

Soil : IRobotSeismicAnalysis_PS_69_SoilType
soil
Available since version 2.5.

Alpha : double
seismic intensity factor
Available since version 2.5.

Delta : double
foundation coefficient
Available since version 2.5.

IRobotSeismicAnalysis_PS_92_Params

Seismic analysis parameters for the French PS 92 code.
Available since version 2.5.

Properties:

Direction : IRobotGeoPoint3D
seismic excitation direction
Available since version 2.5.

Filter : IRobotCaseAnalysisModesFilter
description of modes taken into account during structure dynamic analysis
Available since version 2.5.

ZoneType : IRobotSeismicAnalysis_PS_92_ZoneType
seismic zone number
Available since version 2.5.

Topography : double
topography
Available since version 2.5.

StructureType : IRobotSeismicAnalysis_PS_92_StructureType
structure type
Available since version 2.5.

Site : IRobotSeismicAnalysis_PS_92_SiteType
site
Available since version 2.5.

SpectrumType : IRobotSeismicAnalysisSpectrumType
spectrum
Available since version 2.5.

DirectionType : IRobotSeismicAnalysisDirectionType

direction
Available since version 2.5.

BehaviorFactor : double
behavior factor
Available since version 2.5.

SiteEnvelope : IRobotSeismicAnalysis_PS_92_SiteEnvelope

Available since version 5.5.

ExcitationDir : IRobotDynamicAnalysisExcitationDirection
definition of excitation direction
Available since version 5.5.

IRobotSeismicAnalysis_RPA_88_Params

Seismic analysis parameters for the Algerian RPA 88 code.
Available since version 2.5.

Properties:

Direction : IRobotGeoPoint3D
seismic excitation direction
Available since version 2.5.

Filter : IRobotCaseAnalysisModesFilter
description of modes taken into account during structure dynamic analysis
Available since version 2.5.

ZoneType : IRobotSeismicAnalysis_RPA_88_ZoneType
seismic zone number
Available since version 2.5.

Usage : IRobotSeismicAnalysis_RPA_88_UsageType
usage
Available since version 2.5.

Category : IRobotSeismicAnalysis_RPA_88_CategoryType
category
Available since version 2.5.

Soil : IRobotSeismicAnalysis_RPA_88_SoilType
soil
Available since version 2.5.

QualityFactor : double
quality coefficient
Available since version 2.5.

IRobotSeismicAnalysis_TURKISH_23098_Params

Seismic analysis parameters for the Turkish code.
Available since version 2.5.

Properties:

Direction : IRobotGeoPoint3D
seismic excitation direction
Available since version 2.5.

Filter : IRobotCaseAnalysisModesFilter
description of modes taken into account during structure dynamic analysis
Available since version 2.5.

ZoneType : IRobotSeismicAnalysis_TURKISH_23098_ZoneType
seismic zone number
Available since version 2.5.

SoilType : IRobotSeismicAnalysis_TURKISH_23098_SoilType
soil class
Available since version 2.5.

StructureImportance : double
building importance coefficient
Available since version 2.5.

BehaviorFactor : double
behavior factor
Available since version 2.5.

IRobotSeismicAnalysis_UBC_97_Params

Seismic analysis parameters for the American UBC 97 code.
Available since version 2.5.

Properties:

Direction : IRobotGeoPoint3D
seismic excitation direction
Available since version 2.5.

Filter : IRobotCaseAnalysisModesFilter
description of modes taken into account during structure dynamic analysis
Available since version 2.5.

ZoneType : IRobotSeismicAnalysis_UBC_97_ZoneType
seismic zone number
Available since version 2.5.

Soil : IRobotSeismicAnalysis_UBC_97_SoilType
soil
Available since version 2.5.

ClosestDistance : double
minimum distance to a known seismic source (in km)
Available since version 2.5.

Source : IRobotSeismicAnalysis_UBC_97_SourceType
seismic source type
Available since version 2.5.

ExcitationDir : IRobotDynamicAnalysisExcitationDirection
BehaviorFactor : double
behavior factor

I : double
structure importance factor
Available since version 8.

IRobotSeismicAnalysis_IBC_2000_Params

Seismic analysis parameters for the IBC 2000 code issued in the USA.
Available since version 2.5.

Properties:

Direction : IRobotGeoPoint3D
seismic excitation direction
Available since version 2.5.

Filter : IRobotCaseAnalysisModesFilter
description of modes taken into account during structure dynamic analysis
Available since version 2.5.

SiteClass : IRobotSeismicAnalysis_IBC_2000_SiteClassType
soil
Available since version 2.5.

S1 : double
spectral acceleration for short periods
Available since version 2.5.

Ss : double
spectral acceleration for 1-second period
Available since version 2.5.

ExcitationDir : IRobotDynamicAnalysisExcitationDirection
Ie : double

structure importance factor
Available since version 8.

BehaviorFactor : double
behaviour factor
Available since version 8.

IRobotSeismicAnalysis_EAK_2000_Params

Seismic analysis parameters for the Greek E.A.K. 2000 code.
Available since version 2.5.

Properties:

Direction : IRobotGeoPoint3D
seismic excitation direction
Available since version 2.5.

Filter : IRobotCaseAnalysisModesFilter
description of modes taken into account during structure dynamic analysis
Available since version 2.5.

ZoneType : IRobotSeismicAnalysis_EAK_2000_ZoneType
seismic zone number
Available since version 2.5.

FoundationFactor : double
foundation coefficient
Available since version 2.5.

ImportanceFactor : IRobotSeismicAnalysis_EAK_2000_ImportanceFactorType
structure importance
Available since version 2.5.

GroundCategory : IRobotSeismicAnalysis_EAK_2000_GroundCategoryType
ground category
Available since version 2.5.

DirectionType : IRobotSeismicAnalysisDirectionType
direction
Available since version 2.5.

BehaviorFactor : double
behavior factor
Available since version 2.5.

VerticalFoundationFactor : double
foundation coefficient for the vertical direction
Available since version 2.5.

VerticalBehaviorFactor : double
behavior coefficient for the vertical direction

Available since version 2.5.

IRobotSeismicAnalysis_AFPS_90_ZoneType

Available numbers of seismic zones.
Available since version 2.5.

Properties:

I_SAZT_AFPS_90_IA : = 0

Available since version 2.5.

I_SAZT_AFPS_90_IB : = 1

Available since version 2.5.

I_SAZT_AFPS_90_II : = 2

Available since version 2.5.

I_SAZT_AFPS_90_III : = 3

Available since version 2.5.

IRobotSeismicAnalysis_CIRSOC_103_ZoneType

Available numbers of seismic zones.
Available since version 2.5.

Properties:

I_SAZT_CIRSOC_103_0 : = 0

Available since version 2.5.

I_SAZT_CIRSOC_103_1 : = 1

Available since version 2.5.

I_SAZT_CIRSOC_103_2 : = 2

Available since version 2.5.

I_SAZT_CIRSOC_103_3 : = 3

Available since version 2.5.

I_SAZT_CIRSOC_103_4 : = 4

Available since version 2.5.

IRobotSeismicAnalysis_EC_8_ZoneType

Available numbers of seismic zones.
Available since version 2.5.

Properties:

I_SAZT_EC_8_IA : = 0

Available since version 2.5.

I_SAZT_EC_8_IB : = 1

Available since version 2.5.

I_SAZT_EC_8_II : = 2

Available since version 2.5.

I_SAZT_EC_8_III : = 3

Available since version 2.5.

IRobotSeismicAnalysis_P_100_92_ZoneType

Available numbers of seismic zones.
Available since version 2.5.

Properties:

I_SAZT_P_100_92_A : = 0

Available since version 2.5.

I_SAZT_P_100_92_B : = 1

Available since version 2.5.

I_SAZT_P_100_92_C : = 2

Available since version 2.5.

I_SAZT_P_100_92_D : = 3

Available since version 2.5.

I_SAZT_P_100_92_E : = 4

Available since version 2.5.

I_SAZT_P_100_92_F : = 5

Available since version 2.5.

IRobotSeismicAnalysis_PS_92_ZoneType

Available numbers of seismic zones.
Available since version 2.5.

Properties:

I_SAZT_PS_92_IA : = 0

Available since version 2.5.

I_SAZT_PS_92_IB : = 1

Available since version 2.5.

I_SAZT_PS_92_II : = 2

Available since version 2.5.

I_SAZT_PS_92_III : = 3

Available since version 2.5.

IRobotSeismicAnalysis_RPA_88_ZoneType

Available numbers of seismic zones.
Available since version 2.5.

Properties:

I_SAZT_RPA_88_I : = 0

Available since version 2.5.

I_SAZT_RPA_88_II : = 1

Available since version 2.5.

I_SAZT_RPA_88_III : = 2

Available since version 2.5.

IRobotSeismicAnalysis_UBC_97_ZoneType

Available numbers of seismic zones.
Available since version 2.5.

Properties:

I_SAZT_UBC_97_1 : = 0

Available since version 2.5.

I_SAZT_UBC_97_2A : = 1

Available since version 2.5.

I_SAZT_UBC_97_2B : = 2

Available since version 2.5.

I_SAZT_UBC_97_3 : = 3

Available since version 2.5.

I_SAZT_UBC_97_4 : = 4

Available since version 2.5.

IRobotSeismicAnalysis_EAK_2000_ZoneType

Available numbers of seismic zones.
Available since version 2.5.

Properties:

I_SAZT_EAK_2000_I : = 0

Available since version 2.5.

I_SAZT_EAK_2000_II : = 1

Available since version 2.5.

I_SAZT_EAK_2000_III : = 2

Available since version 2.5.

I_SAZT_EAK_2000_IV : = 3

Available since version 2.5.

IRobotSeismicAnalysis_TURKISH_23098_ZoneType

Available numbers of seismic zones.
Available since version 2.5.

Properties:

I_SAZT_TURKISH_23098_1 : = 0

Available since version 2.5.

I_SAZT_TURKISH_23098_2 : = 1

Available since version 2.5.

I_SAZT_TURKISH_23098_3 : = 2

Available since version 2.5.

I_SAZT_TURKISH_23098_4 : = 3

Available since version 2.5.

IRobotSeismicAnalysis_AFPS_90_StructureType

Available structure types.
Available since version 2.5.

Properties:

I_SAST_AFPS_90_A : = 0

Available since version 2.5.

I_SAST_AFPS_90_B : = 1

Available since version 2.5.

I_SAST_AFPS_90_C : = 2

Available since version 2.5.

IRobotSeismicAnalysis_AFPS_90_SiteType

Available types of structure site.
Available since version 2.5.

Properties:

I_SAST_AFPS_90_S0 : = 0

Available since version 2.5.

I_SAST_AFPS_90_S1 : = 1

Available since version 2.5.

I_SAST_AFPS_90_S2 : = 2

Available since version 2.5.

I_SAST_AFPS_90_S3 : = 3

Available since version 2.5.

IRobotSeismicAnalysisSpectrumType

Available spectrum types.
Available since version 2.5.

Properties:

I_SAST_DIMENSIONING : = 0

Available since version 2.5.

I_SAST_ELASITC : = 1

Available since version 2.5.

IRobotSeismicAnalysis_CHINESE_StructureType

Available structure types.
Available since version 2.5.

Properties:

I_SAST_CHINESE_SPECIAL : = 0

Available since version 2.5.

I_SAST_CHINESE_BUILDINGS : = 1

Available since version 2.5.

I_SAST_CHINESE_BRIDGES : = 2

Available since version 2.5.

I_SAST_CHINESE_HARBOR_BUILDINGS : = 3

Available since version 2.5.

IRobotSeismicAnalysis_CHINESE_SiteType

Available types of structure site.
Available since version 2.5.

Properties:

I_SAST_CHINESE_I : = 0

Available since version 2.5.

I_SAST_CHINESE_II : = 1

Available since version 2.5.

I_SAST_CHINESE_III : = 2

Available since version 2.5.

I_SAST_CHINESE_IV : = 3

Available since version 2.5.

IRobotSeismicAnalysis_CHINESE_IntensityType

Available intensity types.
Available since version 2.5.

Properties:

I_SAIT_CHINESE_6 : = 0

Available since version 2.5.

I_SAIT_CHINESE_7 : = 1

Available since version 2.5.

I_SAIT_CHINESE_8 : = 2

Available since version 2.5.

I_SAIT_CHINESE_9 : = 3

Available since version 2.5.

IRobotSeismicAnalysis_CHINESE_DesignType

Available design standards.
Available since version 2.5.

Properties:

I_SADT_CHINESE_A : = 0

Available since version 2.5.

I_SADT_CHINESE_B : = 1

Available since version 2.5.

IRobotSeismicAnalysis_CHINESE_EarthquakeType

Available types of earthquake distance.
Available since version 2.5.

Properties:

I_SAET_CHINESE_NEAR : = 0

Available since version 2.5.

I_SAET_CHINESE_FAR : = 1

Available since version 2.5.

IRobotSeismicAnalysis_CIRSOC_103_SoilType

Available ground types.
Available since version 2.5.

Properties:

I_SAST_CIRSOC_103_I : = 0

Available since version 2.5.

I_SAST_CIRSOC_103_II : = 1

Available since version 2.5.

I_SAST_CIRSOC_103_III : = 2

Available since version 2.5.

IRobotSeismicAnalysis_CIRSOC_103_StructureType

Available structure types.
Available since version 2.5.

Properties:

I_SAST_CIRSOC_103_A0 : = 0

Available since version 2.5.

I_SAST_CIRSOC_103_A : = 1

Available since version 2.5.

I_SAST_CIRSOC_103_B : = 2

Available since version 2.5.

IRobotSeismicAnalysisDirectionType

Available direction types.
Available since version 2.5.

Properties:

I_SADT_HORIZONTAL : = 0

Available since version 2.5.

I_SADT_VERTICAL : = 1

Available since version 2.5.

IRobotSeismicAnalysis_DM_16_1_96_ProtectionCoeffType

Available values of seismic protection coefficients.
Available since version 2.5.

Properties:

I_SAPCT_DM_16_1_96_1 : = 0

Available since version 2.5.

I_SAPCT_DM_16_1_96_12 : = 1

Available since version 2.5.

I_SAPCT_DM_16_1_96_14 : = 2

Available since version 2.5.

IRobotSeismicAnalysis_EC_8_StructureType

Available structure types.
Available since version 2.5.

Properties:

I_SAST_EC_8_B : = 0

Available since version 2.5.

I_SAST_EC_8_C : = 1

Available since version 2.5.

I_SAST_EC_8_D : = 2

Available since version 2.5.

IRobotSeismicAnalysis_EC_8_SoilType

Available ground types.
Available since version 2.5.

Properties:

I_SAST_EC_8_SOIL_A : = 0

Available since version 2.5.

I_SAST_EC_8_SOIL_B : = 1

Available since version 2.5.

I_SAST_EC_8_SOIL_C : = 2

Available since version 2.5.

IRobotSeismicAnalysis_P_100_92_ImportanceClassType

Available importance classes.
Available since version 2.5.

Properties:

I_SAICT_100_92_I : = 0

Available since version 2.5.

I_SAICT_100_92_II : = 1

Available since version 2.5.

I_SAICT_100_92_III : = 2

Available since version 2.5.

I_SAICT_100_92_IV : = 3

Available since version 2.5.

IRobotSeismicAnalysis_PS_69_DampingType

Available damping types.
Available since version 2.5.

Properties:

I_SADT_PS_69_NORMAL : = 0

Available since version 2.5.

I_SADT_PS_69_AVERAGE : = 1

Available since version 2.5.

I_SADT_PS_69_WEAK : = 2

Available since version 2.5.

IRobotSeismicAnalysis_PS_69_SoilType

Available ground types.
Available since version 2.5.

Properties:

I_SAST_PS_69_RIGID : = 0

Available since version 2.5.

I_SAST_PS_69_FLEXIBLE : = 1

Available since version 2.5.

IRobotSeismicAnalysis_PS_92_StructureType

Available structure types.
Available since version 2.5.

Properties:

I_SAST_PS_92_B : = 0

Available since version 2.5.

I_SAST_PS_92_C : = 1

Available since version 2.5.

I_SAST_PS_92_D : = 2

Available since version 2.5.

IRobotSeismicAnalysis_PS_92_SiteType

Available site types.
Available since version 2.5.

Properties:

I_SAST_PS_92_S0 : = 0

Available since version 2.5.

I_SAST_PS_92_S1 : = 1

Available since version 2.5.

I_SAST_PS_92_S2 : = 2

Available since version 2.5.

I_SAST_PS_92_S3 : = 3

Available since version 2.5.

I_SAST_PS_92_ENVELOPE : = 4

Available since version 5.5.

IRobotSeismicAnalysis_RPA_88_UsageType

Available usage types.
Available since version 2.5.

Properties:

I_SAUT_RPA_88_1 : = 0

Available since version 2.5.

I_SAUT_RPA_88_2 : = 1

Available since version 2.5.

I_SAUT_RPA_88_3 : = 2

Available since version 2.5.

IRobotSeismicAnalysis_RPA_88_CategoryType

Available categories.
Available since version 2.5.

Properties:

I_SACT_RPA_88_1 : = 0

Available since version 2.5.

I_SACT_RPA_88_2 : = 1

Available since version 2.5.

I_SACT_RPA_88_3 : = 2

Available since version 2.5.

I_SACT_RPA_88_4 : = 3

Available since version 2.5.

I_SACT_RPA_88_5 : = 4

Available since version 2.5.

I_SACT_RPA_88_6 : = 5

Available since version 2.5.

I_SACT_RPA_88_7 : = 6

Available since version 2.5.

I_SACT_RPA_88_8 : = 7

Available since version 2.5.

IRobotSeismicAnalysis_RPA_88_SoilType

Available soil types.
Available since version 2.5.

Properties:

I_SAST_RPA_88_RIGID : = 0

Available since version 2.5.

I_SAST_RPA_88_FLEXIBLE : = 1

Available since version 2.5.

IRobotSeismicAnalysis_TURKISH_23098_SoilType

Available soil types.
Available since version 2.5.

Properties:

I_SAST_TURKISH_23098_Z1 : = 0

Available since version 2.5.

I_SAST_TURKISH_23098_Z2 : = 1

Available since version 2.5.

I_SAST_TURKISH_23098_Z3 : = 2

Available since version 2.5.

I_SAST_TURKISH_23098_Z4 : = 3

Available since version 2.5.

IRobotSeismicAnalysis_UBC_97_SoilType

Available soil types.
Available since version 2.5.

Properties:

I_SAST_UBC_97_Sa : = 0

Available since version 2.5.

I_SAST_UBC_97_Sb : = 1

Available since version 2.5.

I_SAST_UBC_97_Sc : = 2

Available since version 2.5.

I_SAST_UBC_97_Sd : = 3

Available since version 2.5.

I_SAST_UBC_97_Se : = 4

Available since version 2.5.

I_SAST_UBC_97_Sf : = 5

Available since version 2.5.

IRobotSeismicAnalysis_UBC_97_SourceType

Available seismic source types.
Available since version 2.5.

Properties:

I_SAST_UBC_97_A : = 0

Available since version 2.5.

I_SAST_UBC_97_B : = 1

Available since version 2.5.

I_SAST_UBC_97_C : = 2

Available since version 2.5.

IRobotSeismicAnalysis_IBC_2000_SiteClassType

Available soil types.
Available since version 2.5.

Properties:

I_SASCT_IBC_2000_A : = 0

Available since version 2.5.

I_SASCT_IBC_2000_B : = 1

Available since version 2.5.

I_SASCT_IBC_2000_C : = 2

Available since version 2.5.

I_SASCT_IBC_2000_D : = 3

Available since version 2.5.

I_SASCT_IBC_2000_E : = 4

Available since version 2.5.

I_SASCT_IBC_2000_F : = 5

Available since version 2.5.

IRobotSeismicAnalysis_EAK_2000_ImportanceFactorType

Available importance factors.
Available since version 2.5.

Properties:

I_SAIFT_EAK_2000_SIGMA1 : = 0

Available since version 2.5.

I_SAIFT_EAK_2000_SIGMA2 : = 1

Available since version 2.5.

I_SAIFT_EAK_2000_SIGMA3 : = 2

Available since version 2.5.

I_SAIFT_EAK_2000_SIGMA4 : = 3

Available since version 2.5.

IRobotSeismicAnalysis_EAK_2000_GroundCategoryType

Available ground categories.
Available since version 2.5.

Properties:

I_SAGCT_EAK_2000_ALPHA : = 0

Available since version 2.5.

I_SAGCT_EAK_2000_BETA : = 1

Available since version 2.5.

I_SAGCT_EAK_2000_GAMMA : = 2

Available since version 2.5.

I_SAGCT_EAK_2000_DELTA : = 3

Available since version 2.5.

IRobotSeismicAnalysis_PS_92_SiteEnvelope

Available since version 5.5.

Operations:

SetActive (_site : IRobotSeismicAnalysis_PS_92_SiteType, _is_active : bool = -1)
Function adds/deletes the indicated site type to/from the envelope.

IsActive (_site : IRobotSeismicAnalysis_PS_92_SiteType) : bool
Function returns the TRUE value if the indicated site type has been considered in the envelope.

IRobotSeismicAnalysis_RPS_2000_ZoneType

Properties:

I_SAZT_RPS_2000_1 : = 0
I_SAZT_RPS_2000_2 : = 1
I_SAZT_RPS_2000_3 : = 2

IRobotSeismicAnalysis_RPS_2000_SiteType

Properties:

I_SAST_RPS_2000_S1 : = 0
I_SAST_RPS_2000_S2 : = 1
I_SAST_RPS_2000_S3 : = 2

IRobotSeismicAnalysis_RPS_2000_StructureClass

Properties:

I_SASC_RPS_2000_I : = 0
I_SASC_RPS_2000_II : = 1

IRobotSeismicAnalysis_RPS_2000_Params

Parameters of the Moroccan seismic code R.P.S. 2000.

Properties:

Zone : IRobotSeismicAnalysis_RPS_2000_ZoneType
Site : IRobotSeismicAnalysis_RPS_2000_SiteType
StructureClass : IRobotSeismicAnalysis_RPS_2000_StructureClass
DirectionType : IRobotSeismicAnalysisDirectionType
BehaviorFactor : double
Filter : IRobotCaseAnalysisModesFilter
ExcitationDir : IRobotDynamicAnalysisExcitationDirection

IRobotSeismicAnalysis_RPA_2003_ZoneType

Properties:

I_SAZT_RPA_2003_I : = 0
I_SAZT_RPA_2003_IIa : = 1

I_SAZT_RPA_2003_IIb : = 2
I_SAZT_RPA_2003_III : = 3

IRobotSeismicAnalysis_RPA_2003_UsageType

Properties:

I_SAUT_RPA_2003_1A : = 0
I_SAUT_RPA_2003_1B : = 1
I_SAUT_RPA_2003_2 : = 2
I_SAUT_RPA_2003_3 : = 3

IRobotSeismicAnalysis_RPA_2003_SiteType

Properties:

I_SAST_RPA_2003_S1 : = 0
I_SAST_RPA_2003_S2 : = 1
I_SAST_RPA_2003_S3 : = 2
I_SAST_RPA_2003_S4 : = 3

IRobotSeismicAnalysis_RPA_2003_Params

Parameters of the Algerian seismic code RPA 99 (2003).

Properties:

Zone : IRobotSeismicAnalysis_RPA_2003_ZoneType
Usage : IRobotSeismicAnalysis_RPA_2003_UsageType
Site : IRobotSeismicAnalysis_RPA_2003_SiteType
BehaviorFactor : double
QualityCoef : double
ExcitationDir : IRobotDynamicAnalysisExcitationDirection
Filter : IRobotCaseAnalysisModesFilter

IRobotSeismicAnalysis_ITALY_ORDINANZA_Params

Seismic analysis parameters for the Italian code Ordinanza 3274.

Properties:

Soil : IRobotSeismicAnalysis_ITALY_ORDINANZA_SoilType
Zone : IRobotSeismicAnalysis_ITALY_ORDINANZA_ZoneType
Spectrum : IRobotSeismicAnalysis_ITALY_ORDINANZA_Spectrum
FactorQ : double
Direction : IRobotSeismicAnalysis_ITALY_ORDINANZA_Direction
ExcitationDir : IRobotDynamicAnalysisExcitationDirection
Filter : IRobotCaseAnalysisModesFilter

IRobotSeismicAnalysis_ITALY_ORDINANZA_SoilType

Soil type according to the Italian code Ordinanza 3274.

Properties:

I_SAST_ITALY_ORDINANZA_A : = 0
I_SAST_ITALY_ORDINANZA_B : = 1
I_SAST_ITALY_ORDINANZA_C : = 2
I_SAST_ITALY_ORDINANZA_D : = 3
I_SAST_ITALY_ORDINANZA_E : = 4

IRobotSeismicAnalysis_ITALY_ORDINANZA_ZoneType

Properties:

I_SAZT_ITALY_ORDINANZA_1 : = 0
I_SAZT_ITALY_ORDINANZA_2 : = 1
I_SAZT_ITALY_ORDINANZA_3 : = 2
I_SAZT_ITALY_ORDINANZA_4 : = 3

IRobotSeismicAnalysis_ITALY_ORDINANZA_Spectrum

Properties:

I_SAS_ITALY_ORDINANZA_DESIGN : = 0
I_SAS_ITALY_ORDINANZA_ELASTIC : = 1

IRobotSeismicAnalysis_ITALY_ORDINANZA_Direction

Properties:

I_SAD_ITALY_ORDINANZA_HORIZONTAL : = 0
I_SAD_ITALY_ORDINANZA_VERTICAL : = 1

IRobotSeismicAnalysis_P_100_2006_Params

Parameters of seismic analysis for Romanian code (P100-1-2006).
Available since version 8.1.

Properties:

SpectrumType : IRobotSeismicAnalysisSpectrumType

Available since version 8.1.

ExcitationDir : IRobotDynamicAnalysisExcitationDirection

Available since version 8.1.

Agg : double

Available since version 8.1.

B0 : double

Available since version 8.1.

Tb : double

Available since version 8.1.

Tc : double

Available since version 8.1.

Td : double

Available since version 8.1.

BehaviorFactor : double

Available since version 8.1.

ImportanceFactor : double

Available since version 8.1.

IRobotSeismicAnalysis_IBC_2006_Params

Available since version 8.1.

Properties:

Direction : IRobotGeoPoint3D

Available since version 8.1.

Filter : IRobotCaseAnalysisModesFilter

Available since version 8.1.

SiteClass : IRobotSeismicAnalysis_IBC_2006_SiteClassType

Available since version 8.1.

S1 : double

Available since version 8.1.

Ss : double

Available since version 8.1.

ExcitationDir : IRobotDynamicAnalysisExcitationDirection

Available since version 8.1.

I : double

Available since version 8.1.

BehaviorFactor : double

Available since version 8.1.

TL : double

Available since version 8.1.

IRobotSeismicAnalysis_IBC_2006_SiteClassType

Available since version 8.1.

Properties:

I_SASCT_IBC_2006_A : = 0

Available since version 8.1.

I_SASCT_IBC_2006_B : = 1

Available since version 8.1.

I_SASCT_IBC_2006_C : = 2

Available since version 8.1.

I_SASCT_IBC_2006_D : = 3

Available since version 8.1.

I_SASCT_IBC_2006_E : = 4

Available since version 8.1.

I_SASCT_IBC_2006_F : = 5

Available since version 8.1.

Spectral analysis parameters

Available since version 2.5.

IRobotSpectralAnalysisParams

Parameters of spectral analysis.
Available since version 2.5.

Properties:

Direction : IRobotGeoPoint3D
spectral excitation direction
Available since version 2.5.

Filter : IRobotCaseAnalysisModesFilter
description of modes taken into account during structure dynamic analysis
Available since version 2.5.

Spectrum : IRobotSpectralAnalysisSpectrum
spectrum definition
Available since version 2.5.

ExcitationDir : IRobotDynamicAnalysisExcitationDirection

IRobotSpectralAnalysisSpectrum

Spectrum definition.
Available since version 2.5.

Properties:

Name : BSTR
spectrum name
Available since version 2.5.

Damping : double
damping
Available since version 2.5.

AbscissaXAxisLogarithmicScale : bool
logarithmic scale on the abscissa axis
Available since version 2.5.

AbscissaXAxis : IRobotSpectralAnalysisAbscissaXAxisType
quantity on the abscissa axis

Available since version 2.5.

OrdinateYAxisLogarithmicScale : bool
logarithmic scale on the ordinate axis
Available since version 2.5.

OrdinateYAxis : IRobotSpectralAnalysisOrdinateYAxisType
quantity on the ordinate axis
Available since version 2.5.

Points : IRobotSpectralAnalysisPointsCollection
points defining spectrum
Available since version 2.5.

Operations:

SaveToFile (_file_path : BSTR)
Function saves the spectrum to a file.
Available since version 2.5.

LoadFromFile (_file_path : BSTR, _spectrum_name : BSTR)
Function reads the spectrum from a file.
Available since version 2.5.

Average (_spectrum_1 : IRobotSpectralAnalysisSpectrum, _spectrum_2 : IRobotSpectralAnalysisSpectrum, _damping : double)
Function calculates the spectrum by averaging the indicated spectra.
Available since version 2.5.

AddFromTimeHistory (_time_history : IRobotSpectralAnalysisPointsCollection, _t_min : double, _t_max : double, _points_count : long, _damping : double)
To the collection of points defining spectrum the function adds points of the spectrum calculated based on the time-dependent function and specified parameters.
Available since version 2.5.

IRobotSpectralAnalysisPointsCollection

Collection of points defining spectrum.
Available since version 2.5.

Properties:

Count : long
number of points defining spectrum
Available since version 2.5.

Operations:

Add (_x : double, _y : double)

Function adds a new point to the spectrum definition.
Available since version 2.5.

Set (_pos : long, _x : double, _y : double)
Function modifies the selected point defining the spectrum.
Available since version 2.5.

Get (_pos : long, _x : double*, _y : double*)
Function returns coordinates of the selected point defining spectrum.
Available since version 2.5.

Remove (_pos : long)
Function deletes the point from the collection defining spectrum.
Available since version 2.5.

LoadFromFile (_file_path : BSTR)
Function reads out the point collection from a file.
Available since version 2.5.

SaveToFile (_file_path : BSTR)
Function saves the point collection to a file.
Available since version 2.5.

Clear ()
Function deletes all the points from the collection defining spectrum.
Available since version 2.5.

IRobotSpectralAnalysisAbscissaXAxisType

Available quantities on the abscissa axis.
Available since version 2.5.

Properties:

I_SAAXAT_PERIOD : = 0
period
Available since version 2.5.

I_SAAXAT_PULSATION : = 1
pulsation
Available since version 2.5.

I_SAAXAT_FREQUENCY : = 2
frequency
Available since version 2.5.

IRobotSpectralAnalysisOrdinateYAxisType

Available quantities on the ordinate axis.

Available since version 2.5.

Properties:

I_SAOYAT_VELOCITY : = 0
velocity
Available since version 2.5.

I_SAOYAT_ACCELERATION : = 1
acceleration
Available since version 2.5.

I_SAOYAT_EXCITATION : = 2
displacement
Available since version 2.5.

Non-linear analysis parameters

Available since version 2.5.

IRobotNonlinearAnalysisAlgorithmType

Methods of non-linear analysis.
Available since version 2.5.

Properties:

I_NAAT_INCREMENTAL_METHOD : = 0
incremental method of non-linear analysis
Available since version 2.5.

I_NAAT_DIRECT_ITERATION_METHOD : = 1
direct iteration method
Available since version 2.5.

I_NAAT_ARC_LENGTH_METHOD : = 2

Available since version 2.5.

I_NAAT_PREDICTOR_CORRECTOR_METHOD : = 3

Available since version 3.

IRobotNonlinearAnalysisParams

Parameters of non-linear analysis.
Available since version 2.5.

Properties:

PDelta : bool
definition of P-delta analysis

Algorithm : IRobotNonlinearAnalysisAlgorithmType
method of non-linear analysis

LoadIncrementNumber : long
number of load increments

MaximumIterationNumberForOneIncrement : long
maximum iteration number for one increment

IncrementLengthReductionNumber : long
increment length reduction number

IncrementLengthReductionFactor : double
increment length reduction factor

MaximumNumberOfLineSearches : long
maximum number of line searches

LineSearchMethodFactor : double
control parameter for line-search method

MaximumNumberOfBFGSCorrections : long
maximum number of BFGS corrections

MatrixUpdateAfterEachSubdivision : bool
matrix update after each subdivision

MatrixUpdateAfterEachIteration : bool
matrix update after each iteration

ResultListEachIteration : bool
relative code tolerance for residual forces

ResidualForcesRelativeCodeTolerance : double
relative code tolerance for residual forces

DisplacementsRelativeCodeTolerance : double
relative code tolerance for displacements
Available since version 2.5.

MaxLoadFactor : double
maximum load factor (for the "Arc-length" method)
Available since version 2.5.

NodeNumber : long
node number (for the "Arc-length" method)

Available since version 2.5.

DegreeOfFreedom : IRobotDegreeOfFreedom
definition of degree of freedom (for the "Arc-length" method)
Available since version 2.5.

MaxDisplacement : double
maximum displacement for the selected degree of freedom (for the "Arc-length" method)
Available since version 2.5.

Stiff : bool
(for "Arc-length" method)
Available since version 2.5.

Operations:

GetSettingsFromPreferences ()
Function takes settings from preferences.
Available since version 2.5.

SaveSettingsInPreferences ()
Function saves settings to preferences as default ones.
Available since version 2.5.

Buckling analysis parameters

Available since version 3.

IRobotBucklingAnalysisParams

Buckling analysis parameters
Available since version 3.

Properties:

ModesCount : long
Number of buckling modes
Available since version 3.

Tolerance : double
value of the error tolerance that is to be obtained during structure iteration analysis
Available since version 3.

IterationsCount : long
Maximum number of iterations
Available since version 3.

Shift : double
Shift value
Available since version 3.

Increment : long
Number of load increments
Available since version 3.

Method : IRobotBucklingAnalysisMethod
method of eigenproblem solution
Available since version 3.

SturmVerification : bool
flag switching on/off the algorithm that enables searching disregarded structure modes
Available since version 3.

IsNonlinear : bool

Available since version 3.

NonlinearParams : IRobotNonlinearAnalysisParams
Parameters of non-linear analysis (available if IsNonlinear flag is set)
Available since version 3.

IRobotBucklingAnalysisMethod

Methods of eigenproblem solution for buckling analysis.
Available since version 3.

Properties:

I_BAM_SUBSPACE_ITERATION : = 0
Subspace iteration
Available since version 3.

I_BAM_BLOCK_SUBSPACE_ITERATION : = 1
Block subspace iteration
Available since version 3.

Time history analysis parameters

Available since version 3.

IRobotTimeHistoryAnalysisParams

Parameters of time history analysis
Available since version 3.

Properties:

Method : IRobotTimeHistoryAnalysisMethod
Time history analysis method
Available since version 3.

MethodParams : IDispatch
Parameters of selected method
Available since version 3.

TimeStep : double
step of the time variable for which results are being saved
Available since version 3.

Division : long
number of time division for the result time step
Available since version 3.

End : double
final value of the time variable for which the analysis is performed
Available since version 3.

Count : long
number of defined time history analyses
Available since version 3.

Operations:

Set (_case_num : long, _time_fun : string, _coeff : double, _shift : double)
Function defines the time history analysis for the specified static case or mass.
Available since version 3.

Find (_case_num : long) : long
Function returns index of the time history analysis definition for the specified static case (or mass). If for the indicated case the time history analysis has not been defined, function returns zero value.
Available since version 3.

Delete (_idx : long)
Function deletes time history analysis definition of the index specified on the list.
Available since version 3.

Get (_idx : long, _case_num : long*, _time_fun : string*, _coeff : double*, _shift : double*)
Function returns the time history analysis definition of the specified index.
Available since version 3.

IRobotTimeHistoryAnalysisMethod

Available methods of time history analysis.
Available since version 3.

Properties:

I_THAM_NEWMARK : = 0

Available since version 3.

I_THAM_MODAL_DECOMPOSITION : = 1

Available since version 3.

I_THAM_PREDICTOR_CORECTOR : = 2

Available since version 3.

IRobotTimeHistoryNewmarkParams

Parameters of Newmark method for time history analysis.
Available since version 3.

Properties:

MassMatrixType : IRobotModalAnalysisMassMatrixType
Stiffness matrix type
Available since version 3.

Alpha : double
Damping parameter
Available since version 3.

Beta : double
Damping parameter
Available since version 3.

IRobotTimeHistoryModalDecompositionParams

Parameters of modal decomposition method.
It is a list of pairs [mode number, damping value]. For each mode, maximally one such a pair may exist. Additionally, the pairs are indexed with numbers from 1 to Count.
Available since version 3.

Properties:

Count : long

number of defined mode pairs [mode number, damping value]
Available since version 3.

Operations:

Get (_idx : long, _mode : long*, _damp : double*)
Function returns the next pair from the list.
Available since version 3.

Delete (_idx : long)
Function removes the selected list element.
Available since version 3.

SetDamping (_mode : long, _damp : double)
Function sets a damping value for a given mode.
Available since version 3.

GetDamping (_mode : long) : double
Function returns a damping value for a selected mode.
Available since version 3.

IsDefined (_mode : long) : bool
Function returns non-zero value (True), if for the specified mode damping has been defined.
Available since version 3.

DeleteMode (_mode : long)
Function deletes the pair containing the selected mode from the list.
Available since version 3.

IRobotTimeHistoryPointsCollection

Collection of points defining a time function for the time history analysis.
Available since version 3.

Properties:

Count : long
Number of points in a collection
Available since version 3.

Operations:

Add (_x : double, _y : double)
Function adds a new point of the (X, Y) coordinates. If a point of X coordinate has been defined earlier, then its value will be replaced by a new Y value.
Available since version 3.

Clear ()
Function removes all collection points.

Available since version 3.

Find (_x : double) : long
Function returns index of the point defined for the X coordinate. If for the indicated X coordinate, Y value has not been defined, then function returns zero value (0).
Available since version 3.

Get (_idx : long, _x : double*, _y : double*)
Function returns selected collection point.
Available since version 3.

Delete (_idx : long)
Function removes selected collection point.
Available since version 3.

LoadFromFile (_file_path : string)
Function reads in the collection of points from the file and adds it to the points currently defined.
Available since version 3.

SaveToFile (_file_path : string)
Function saves the collection of points to a file.
Available since version 3.

IRobotTimeHistoryFunctionList

List of defined time functions for time history analysis.
Available since version 3.

Properties:

Count : long
Number of defined (available) time functions
Available since version 3.

Operations:

Get (_idx : long) : IRobotTimeHistoryPointsCollection
Function returns the collection of points defining a time function of the indicated index.
Available since version 3.

Delete (_idx : long)
Function deletes the definition of time function of the specified index.
Available since version 3.

Create () : IRobotTimeHistoryPointsCollection
Function creates and returns a new (empty) collection of points defining a time function.
Available since version 3.

Store (_function_name : string, _points : IRobotTimeHistoryPointsCollection) : long
Function saves the indicated collection of points as a definition of time function of the specified name. Index of the function definition from the list is returned.
Available since version 3.

Find (_function_name : string) : long
Function returns index of the time function definition of the indicated name.
Available since version 3.

GetName (_idx : long) : string
Function returns a name of the time function of the indicated index.
Available since version 3.

CreateSum (_fun1_idx : long, _fun1_coeff : double, _fun2_idx : long, _fun2_coeff : double) : IRobotTimeHistoryPointsCollection
Function creates and returns the collection of points as a sum of the indicated time functions.
Available since version 3.

AddFromFile (_file_path : string)
Function reads in the list of time functions from the indicated file.
Available since version 3.

SaveToFile (_file_path : string)
Function saves the list of time functions to the indicated file.
Available since version 3.

IRobotTimeHistoryNonlinearParams

Non-linear parameters of time history analysis
Available since version 3.

Properties:

PDelta : bool
Definition of P-Delta analysis
Available since version 3.

MaximumIterationNumberForOneIncrement : long
maximum number of iterations for an increment
Available since version 3.

ResidualForcesRelativeCodeTolerance : double
Relative norm tolerance for residual forces
Available since version 3.

MatrixUpdateAfterEachSubdivision : bool
matrices are updated after each subdivision
Available since version 3.

MatrixUpdateAfterEachIteration : bool
Matrix update after each iteration
Available since version 3.

IRobotTimeHistoryPredictorCorectorParams

Parameters of "predictor-corector" method for time history analysis.
Available since version 3.

Properties:

Beta : double

Available since version 3.

NonlinearParams : IRobotTimeHistoryNonlinearParams
Non-linear parameters
Available since version 3.

Harmonic analysis parameters

Available since version 3.

IRobotHarmonicAnalysisParams

Available since version 3.

Properties:

MassMatrix : IRobotModalAnalysisMassMatrixType
Stiffness matrix type
Available since version 3.

Excitation : IRobotModalAnalysisLimits
excitation
Available since version 3.

Push-over analysis parameters

Available since version 3.

IRobotPushOverAnalysisParams

Parameters of Push-over analysis
Available since version 3.

Properties:

Node : long
Node number
Available since version 3.

Direction : IRobotPushOverDirection
Direction
Available since version 3.

MaxDisplacement : double
maximum displacement
Available since version 3.

LoadDefinition : IRobotPushOverLoadDefinitionMethod
Method of load definition
Available since version 3.

Nonlinear : bool
flag switching on/off a non-linear analysis
Available since version 3.

NonlinearParams : IRobotNonlinearAnalysisParams
non-linear analysis parameters (considered only if the Nonlinear flag is set)
Available since version 3.

IRobotPushOverLoadDefinitionMethod

Available methods of load definition for push-over analysis.
Available since version 3.

Properties:

I_POLDM_ACCELERATION : = 0
load definition based on the acceleration in the indicated direction
Available since version 3.

I_POLDM_USER_DEFINED : = 2

Available since version 3.

IRobotPushOverDirection

Available directions for Push-over analysis

Available since version 3.

Properties:

I_POD_UX_PLUS : = 0

Available since version 3.

I_POD_UX_MINUS : = 10

Available since version 3.

I_POD_UY_PLUS : = 1

Available since version 3.

I_POD_UY_MINUS : = 11

Available since version 3.

I_POD_UZ_PLUS : = 2

Available since version 3.

I_POD_UZ_MINUS : = 12

Available since version 3.

Moving loads

Available since version 2.5.

IRobotMobileCase

Derived from IRobotCase
Moving load case.
Available since version 2.5.

Properties:

Components : IRobotMobileCaseComponentMngr
list of moving load case components; Each component corresponds to a simple case generated due to applying the vehicle to the determined node on the route.
Available since version 2.5.

Flag : IRobotMobileCaseFlag

flag determining type of a moving load case; Possible flag values are described by the IRobotMobileCaseFlag interface; In practice, this field is used for navigation between the base moving load case and its envelopes.
Available since version 2.5.

UniqueId : long
unique identifier of a load case
Available since version 2.5.

ApplicationPlaneType : IRobotMobileCaseApplicationPlaneType
type of the plane to which the vehicle is applied
Available since version 3.

Vehicle : string
name identifying the vehicle
Available since version 3.

ApplicationPlaneBars : string
list of bars defining the plane to which the vehicle is applied (active parameter for ApplicationPlaneType = I_MCAPT_SELECTION)
Available since version 3.

NatureName : string
name of nature
Available since version 3.

Label : string

Available since version 8.

Operations:

FindByFlag (_mc_flag : IRobotMobileCaseFlag) : long
Function returns number of the case of the determined type, associated with this moving load case. It allows navigation between the base moving load case and its envelopes.

Available since version 2.5.

GetRoute () : IRobotMobileCaseRoute
Function takes the interface of route definition.
Available since version 3.

SetRoute (_route : IRobotMobileCaseRoute)
Function updates the route definition based on the indicated interface.
Available since version 3.

SetNatureExt (_rgl_action : long)
Function assigns - to the case - the nature consistent with the indicated number of "action" described in the current regulations.

Available since version 3.

IsAuxiliary : bool

Available since version 8.1.

IRobotMobileCaseComponentMngr

Component list of moving load.
Available since version 2.5.

Properties:

Count : long
number of components on the list
Available since version 2.5.

Operations:

Get (_cmpnt_idx : long) : IRobotMobileCaseComponent
Function returns the component with the specified index. Components of a moving load case are indexed from 1 to Count.
Available since version 2.5.

IRobotMobileCaseComponent

Simple component of a moving load case. This component corresponds to the simple case generated at the next point of the route.
Available since version 2.5.

Properties:

Point : long
number of the point on the route
Available since version 2.5.

Records : IRobotLoadRecordMngr
list of load records
Available since version 2.5.

IRobotMobileCaseFlag

Available types of a moving load case.
Available since version 2.5.

Properties:

I_MCF_MAIN : = 0

base moving load case
Available since version 2.5.

I_MCF_MAX : = 1
top envelope
Available since version 2.5.

I_MCF_MIN : = 2
bottom envelope
Available since version 2.5.

IRobotMobileCaseRoute

Definition of vehicle route
Available since version 3.

Properties:

Geometry : long
number of the object that determines route geometry
Available since version 3.

Step : double
step - the distance between two successive vehicle positions on a route, expressed in meters
Available since version 3.

LoadDirection : IRobotGeoPoint3D
Global direction of load application
Available since version 3.

EndRouteLimit : bool
Limit of vehicle position at the route end
Available since version 3.

BeginingRouteLimit : bool
Limit of vehicle position at the route beginning
Available since version 3.

Tolerance : double
value determining conditions of generating the loads associated with the vehicle in the indicated structure
Available since version 3.

Operations:

GetFactors (_segment : long) : IRobotMobileCaseSegmentFactors
Function takes the vector of coefficients correcting loads for a route segment of the indicated number
Available since version 3.

SetFactors (_segment : long, _factors : IRobotMobileCaseSegmentFactors)
Function updates the vector of coefficients correcting loads for a route segment of the indicated number and on the basis of the indicated interface

Available since version 3.

IRobotMobileCaseSegmentFactors

A set of coefficients assigned to a route segment, correcting vehicle loads.
Available since version 3.

Properties:

Gamma : double
angle determining vehicle rotation about the vertical axis
Available since version 3.

VR : double
Scaling factor for vertical force V (from right side)
Available since version 3.

VL : double
Scaling factor for vertical force V (from left side)
Available since version 3.

HR : double
Scaling factor for horizontal transversal force H (from right side)
Available since version 3.

HL : double
Scaling factor for horizontal transversal force H (from left side)
Available since version 3.

LR : double
Scaling factor for horizontal longitudinal force L (from right side)
Available since version 3.

LL : double
Scaling factor for horizontal longitudinal force L (from left side)
Available since version 3.

IRobotMobileCaseApplicationPlaneType

Type of application plane.
Available since version 3.

Properties:

I_MCAPT_AUTOMATIC : = 1
Application - automatic
Available since version 3.

I_MCAPT_SELECTION : = 2
Application - to selected bars
Available since version 3.

IRobotVehicleData

Parameter set for vehicle definition
Available since version 3.

Properties:

b : double
Vehicle width
Available since version 3.

d1 : double
distance between load and vehicle contour at the vehicle front
Available since version 3.

d2 : double
distance between load and vehicle contour at the vehicle back
Available since version 3.

Loads : IRobotVehicleLoadMngr
list of loads associated with the vehicle
Available since version 3.

Operations:

LoadFromDBase (_vehicle_name : string, _db_name : string) : bool
Function reads vehicle parameters from the database.
Available since version 3.

StoreToDBase (_vehicle_name : string, _db_name : string) : bool
Function saves vehicle parameters to the database
Available since version 3.

IRobotVehicleLoadMngr

List of loads that define a vehicle
Available since version 3.

Properties:

Count : long

Number of records on the list
Available since version 3.

Operations:

Get (_idx : long) : IRobotVehicleLoad
Function takes a record of the indicated index.
Available since version 3.

Delete (_idx : long)
Function removes a record from a list
Available since version 3.

New () : IRobotVehicleLoad
Function adds a new load record at the end of the list.
Available since version 3.

IRobotVehicleLoad

Record describing a single load associated with the vehicle.
Available since version 3.

Properties:

F : double
depending on a load type, it is: value of a concentrated force, value of a linear load or value of a planar load
Available since version 3.

X : double
coordinate value for the line along which the force is applied along the vehicle axis
Available since version 3.

S : double
depending on a load type, it is: value of force spacing, value of linear load spacing (only in the direction of Y axis) or value of planar load spacing (only in the direction of Y axis)
Available since version 3.

DX : double
for a linear load, it is the length of a segment on which the load is acting (along the vehicle axis);
for a planar load, it is the length of a rectangle side on which the load is acting (along the vehicle axis);
for concentrated force loads, the parameter is inactive
Available since version 3.

DY : double
for a linear load, it is the length of a segment on which the load is acting (perpendicular to the vehicle axis);

for a planar load, it is the length of a rectangle side on which the load is acting (perpendicular to the vehicle axis);
for concentrated force loads, the parameter is inactive

Available since version 3.

Type : IRobotVehicleLoadType
Load type
Available since version 3.

IRobotVehicleLoadType

Load types
Available since version 3.

Properties:

I_VLT_CONCENTRATED : = 1
Concentrated force
Available since version 3.

I_VLT_LINEAR : = 2
Linear load
Available since version 3.

I_VLT_SURFACE : = 3
Surface load
Available since version 3.

IRobotVehicleDatabaseList

Interface describing the list of vehicle databases.
Available since version 3.

Properties:

Count : int
Number of vehicle databases on the list
Available since version 3.

Default : int
Index of default vehicle database
Available since version 3.

Operations:

Add (_db_name : string) : bool
Function adds vehicle database of the indicated name at the end of the list.
Status of the action is returned.

Available since version 3.

Find (_db_name : string) : int
Function searches the list seeking for the database of the indicated name.
If there is no specified database on the list, zero value is returned, if there is, function returns the database index from the list.
Available since version 3.

GetDatabase (_idx : int) : IRobotVehicleDatabase
Function returns the interface for the vehicle database with the index indicated on the list.
Available since version 3.

Get (_idx : int) : String
Function returns the name of the vehicle database located at the indicated position on the list.
Available since version 3.

Remove (_idx : int)
Function deletes the database of the indicated index from the list.
Available since version 3.

AddFromFile (_file_path : string) : bool
Function adds the vehicle database saved in the indicated file at the end of the list. Status of the operation execution is returned.
Available since version 3.

Create (_db_name : string, _long_db_name : string, _description : string)
Function creates a new vehicle database.
Available since version 3.

IRobotVehicleDatabase

Interface for vehicle database support.
Available since version 3.

Properties:

Name : string
name identifying the vehicle database
Available since version 3.

LongName : string
Additional description of a vehicle database
Available since version 3.

Description : string

Available since version 3.

Operations:

Load (_db_name : string) : bool
Function reads vehicles from the vehicle database of the indicated name. If the specified database does not exist or an error occurs during read-out, zero value is returned.
Available since version 3.

GetAll () : IRobotNamesArray
Function returns a table containing names of all the vehicles from the database.
Available since version 3.

LoadFromFile (_file_path : string) : bool
Function reads vehicles from the database saved in the indicated file. If the specified path is incorrect or other reading error occurs, the function returns zero value.
Available since version 3.

Code combinations

IRobotCodeCombinationEngine

Properties:

Params : IRobotCodeCmbGenerationParams
parameters of code combination generation
Available since version 1.7.

Operations:

Generate ()
Function generates code-defined loads according to the selected code and current settings of generation parameters.
Available since version 1.7.

IRobotCodeCombination

Derived from IRobotCase
Combination of load cases defined according to selected code.

Properties:

UniqueId : long
unique combination identifier
Available since version 1.7.

CombinationType : IRobotCombinationType
combination type

Available since version 1.7.

Flag : IRobotCodeCmbFlag
Available since version 1.7.

Components : IRobotCodeCmbComponentMngr
set of component combinations
Available since version 1.7.

Operations:

FindByFlag (cmb_flag : IRobotCodeCmbFlag) : long
Function returns a user's number for the code combination linked with the combination that has a determined flag. If such a combination is not found, then zero value is returned.
Available since version 1.7.

IRobotCodeCmbComponentMngr

Access interface to definitions of code combination components.

Properties:

Count : long
number of combinations
Available since version 1.7.

Operations:

Get (cmb_idx : long) : IRobotCodeCmbComponent
Function returns a set of pairs (case number, coefficient) defining the selected combination.
Available since version 1.7.

IRobotCodeCmbFactor

Properties:

CaseNumber : long
Available since version 1.7.

Factor : double
Available since version 1.7.

IRobotCodeCmbComponent

Properties:

Count : long
Available since version 1.7.

Operations:

Get (idx : long) : IRobotCodeCmbFactor
Available since version 1.7.

IRobotCodeCmbFlag

Properties:

I_CCF_MAIN : = 3
Available since version 1.7.

I_CCF_MAX : = 0
Available since version 1.7.

I_CCF_MIN : = 1
Available since version 1.7.

IRobotCodeCmbRegulations

Definition of the regulations describing the manner of generating code-defined loads.

Properties:

IsEuroCode : bool
flag indicating if the regulations are compatible with the EUROCODE standard
Available since version 1.7.

CodeName : string
name of the regulations
Available since version 1.7.

MaterialType : IRobotMaterialType
material type;
material type is recognized by its name - in case when a non-standard material name is used, its type will not be recognized correctly and the value designating "OTHER" unknown material type will be returned
Available since version 1.7.

Version : string
text describing the version of the regulations
Available since version 1.7.

Actions : IRobotCodeCmbActionServer
actions described in the regulations
Available since version 1.7.

Combinations : IRobotCodeCmbCombs
combinations described in the regulations
Available since version 1.7.

IRobotCodeCmbActionServer

Definition of actions.

Properties:

Count : long
number of defined actions
Available since version 1.7.

Operations:

SetCoeff (idx : long, coeff_type : IRobotCodeCmbActionCoeffType, coeff_val : double)
Function sets the value of a specified coefficient for a specified action.
Available since version 1.7.

GetCoeff (idx : long, coeff_type : IRobotCodeCmbActionCoeffType) : double
Function returns the value of a selected coefficient for a selected action.
Available since version 1.7.

New (case_nature : IRobotCaseNature, _name : string) : long
Function defines a new action and returns its index which should be used while determining the values of individual coefficients.
Available since version 1.7.

Remove (idx : long)
Function deletes the indicated action definition.
Available since version 1.7.

GetNature (idx : long) : IRobotCaseNature
Function returns the nature of the selected action
Available since version 1.7.

SetNature (idx : long, case_nat : IRobotCaseNature)
Function changes the nature of a selected action.
Available since version 1.7.

GetName (idx : long) : string
Function returns the name of the indicated action.
Available since version 1.7.

SetName (idx : long, _name : string)

Function sets (changes) the name of the indicated action.
Available since version 1.7.

IRobotCodeCmbActionCoeffType

Coefficients used for describing actions.

Properties:

I_CCACT_GU_MAX : = 0
partial safety factor for ULS (maximum)
Available since version 1.7.

I_CCACT_GU_MIN : = 1
partial safety factor for ULS (minimum)
Available since version 1.7.

I_CCACT_GS : = 2
partial safety factor for SLS
Available since version 1.7.

I_CCACT_GA : = 3
partial safety factor for accidental state
Available since version 1.7.

I_CCACT_PSI0_1 : = 4
simultaneity coefficient for the first load
Available since version 1.7.

I_CCACT_PSI0_2 : = 5
simultaneity coefficient for the second load
Available since version 1.7.

I_CCACT_PSI0_3 : = 6
simultaneity coefficient for the third load
Available since version 1.7.

I_CCACT_PSI0_N : = 7
simultaneity coefficient for the fourth load
Available since version 1.7.

I_CCACT_PSI1 : = 8
factor for long-term loads
Available since version 1.7.

I_CCACT_PSI2_1 : = 9
reduction factor for accidental combination
Available since version 1.7.

I_CCACT_PSI2_N : = 10

reduction factor
Available since version 1.7.

I_CCACT_PSIK : = 11
coefficient for second load and next loads
Available since version 1.7.

I_CCACT_KSI_MAX : = 12
Available since version 1.7.

I_CCACT_KSI_MIN : = 13
Available since version 1.7.

IRobotCodeCmbCombs

Combination definition.

Properties:

Count : long
number of combinations
Available since version 1.7.

Operations:

New (lim_state : IRobotLimitState, cmb_type : IRobotCodeCmbCombType, _name : string, dead_coeff : short, live_coeff : short, acc_coeff : short, seis_coeff : short) : long
Function creates a new combination and adds its definition at the end of the list. Index of the newly-created combination is returned.
Available since version 1.7.

Get (idx : long, limit_state : IRobotLimitState*, cmb_type : IRobotCodeCmbCombType*, _name : string*, dead_coef : short*, live_coef : short*, acc_coef : short*, seis_coef : short*)
Function returns the definition of selected combination.
Available since version 1.7.

Set (idx : long, limit_state : IRobotLimitState, cmb_type : IRobotCodeCmbCombType, _name : string, dead_coef : short, live_coef : short, acc_coef : short, seis_coef : short)
Function changes the definition of the selected combination.
Available since version 1.7.

Remove (idx : long)
Function deletes the definition of the selected combination.
Available since version 1.7.

IRobotCodeCmbCombPartType

Properties:

I_CCCPT_DEAD : = 0
Available since version 1.7.

I_CCCPT_LIVE : = 1
Available since version 1.7.

I_CCCPT_ACCIDENTAL : = 2
Available since version 1.7.

I_CCCPT_SEISMIC : = 3
Available since version 1.7.

IRobotCodeCmbCombType

Properties:

I_CCCT_STANDARD : = 1
standard combination
Available since version 1.7.

I_CCCT_SEISMIC : = 2
seismic combination
Available since version 1.7.

I_CCCT_ACCIDENTAL : = 3
accidental combination
Available since version 1.7.

I_CCCT_RARE : = 4
rare combination
Available since version 1.7.

I_CCCT_FREQUENT : = 5
frequent combination
Available since version 1.7.

I_CCCT_QUASI_PERM : = 6
quasi-permanent combination
Available since version 1.7.

I_CCCT_USER : = 7
user-defined combination
Available since version 1.7.

I_CCCT_FUNDAMENTAL : = 8
basic combination
Available since version 1.7.

I_CCCT_SIMPLIFIED : = 9
simplified combination
Available since version 1.7.

I_CCCT_EXTREMAL : = 10
extreme combination
Available since version 1.7.

IRobotCodeCmbGenerationParams

Parameters of code combination generation

Properties:

AllNodes : bool
Available since version 1.7.

NodeSel : string
node selection (taken into account only for simplified generation when AllNodes are set as zero - False)
Available since version 1.7.

AllBars : bool
Available since version 1.7.

BarSel : string
bar selection (taken into account only for simplified generation when AllBars are set as zero - False)
Available since version 1.7.

Simplified : bool
flag imposing simplifed generation
Available since version 1.7.

PointsOnBar : short
number of points along the bar length (it is taken into account only for simplified generation)
Available since version 1.7.

ExtremalSnowFactor : double
factor for extreme snow
Available since version 1.7.

Regulations : IRobotCodeCmbRegulations
definition of the regulations
Available since version 1.7.

Groups : IRobotCodeCmbGroupServer
groups
Available since version 1.7.

Relations : IRobotCodeCmbGroupRelationServer
relations between groups
Available since version 1.7.

ActiveCases : IRobotCollection
information about cases participating in a definition of code combinations (object collection of the RobotCodeCmbActiveCaseInfo type)
Available since version 1.7.

Operations:

SelectDecidingValue (dvalue : IRobotCodeCmbDecidingValueType, _select : bool)
Function enables selection of decisive values.
Available since version 1.7.

IsDecidingValueSelected (dvalue : IRobotCodeCmbDecidingValueType) : bool
Function returns a value different from zero (True) if the specified decisive value has been selected.
Available since version 1.7.

SelectCombination (cmb_idx : long, _select : bool)
Function enables selection of templates for calculating a combination among all the templates defined in the regulations.
Available since version 1.7.

IsCombinationSelected (cmd_idx : long) : bool
Function returns a value different from zero (True) if a specified template for generating combination is to be taken into account during generation.
Available since version 1.7.

IRobotCodeCmbDecidingValueType

Properties:

I_CCDVT_FX : = 1
Available since version 1.7.

I_CCDVT_FY : = 2
Available since version 1.7.

I_CCDVT_FZ : = 3
Available since version 1.7.

I_CCDVT_MX : = 4

Available since version 1.7.

I_CCDVT_MY : = 5
Available since version 1.7.

I_CCDVT_MZ : = 6
Available since version 1.7.

I_CCDVT_UX : = 7
Available since version 1.7.

I_CCDVT_UY : = 8
Available since version 1.7.

I_CCDVT_UZ : = 9
Available since version 1.7.

I_CCDVT_FXX_MAX : = 10
Available since version 1.7.

I_CCDVT_MXX_MAX : = 11
Available since version 1.7.

I_CCDVT_SXX_MAX : = 12
Available since version 1.7.

I_CCDVT_REACTIONS : = 35
Available since version 1.7.

I_CCDVT_SIGMA_X : = 13
Available since version 1.7.

I_CCDVT_DEFLECTION : = 14
Available since version 1.7.

IRobotCodeCmbGroup

Definition of load case groups.

Properties:

Nature : IRobotCaseNature
nature of cases included in the group
Available since version 1.7.

Operator : IRobotCodeCmbOperator
operator determining relation between the group components
Available since version 1.7.

CaseCount : long

number of load cases included in the group
Available since version 1.7.

Operations:

Add (case_num : long) : bool
Function adds a case with the specified number to the group. If the case nature is incompatible with the group nature the function returns zero (False).
Adding a case to this group means that it is simultaneously deleted from the group to which it has hitherto belonged.
Available since version 1.7.

AddAll ()
Functions adds to the group all the load cases whose nature is compatible with the group nature. Simultaneously. it means that these cases will be deleted from the groups to which they have hitherto belonged.
Available since version 1.7.

IsFull () : bool
Function returns a value different from zero (True) if the natures of all the load cases in a group are compatible with the group nature.
Available since version 1.7.

IRobotCodeCmbOperator

Available operators used for defining relations between cases and case groups.

Properties:

I_CCO_AND : = 0
operator: and
Available since version 1.7.

I_CCO_EXCLUSIVE_OR : = 1
operator: exclusive or
Available since version 1.7.

I_CCO_AND_OR : = 2
operator: and/or
Available since version 1.7.

IRobotCodeCmbGroupServer

Server managing the groups of cases

Operations:

New (grp_nature : IRobotCaseNature, oper : IRobotCodeCmbOperator, case_list : string) : long

Function creates a new group with the defined parameters and returns its number. Groups are numbered within a given nature from 1 to Count.
Available since version 1.7.

Get (grp_nature : IRobotCaseNature, grp_number : long) : IRobotCodeCmbGroup
Function returns a group with the specified number among the groups of the defined nature. Groups are numbered within the nature from 1 to their number.
Available since version 1.7.

FindByCase (case_num : long) : long
Function finds the group which includes the specified load case and returns its number among the groups whose nature is compatible with the nature of that case.
Available since version 1.7.

GetCount (grp_nature : IRobotCaseNature) : long
Function returns the number of groups of the specified nature.
Available since version 2.0.

IRobotCodeCmbGroupRelation

Definition of relations between groups.
Definition of relations between groups can be illustrated by means of the matrix whose elements are group numbers. All groups whose numbers are located in the same matrix row are in the same relation with each other (either AND or AND_OR). Whereas among the matrix rows the relation EXCLUSIVE_OR occurs.

Properties:

Nature : IRobotCaseNature
nature of groups defining relation
Available since version 1.7.

RowCount : long
number of rows defining relation
Available since version 1.7.

Operations:

GetGroupCount (row_idx : long) : long
Function returns the number of groups defining a selected matrix row. Matrix rows are numbered from 1 to RowCount.
Available since version 1.7.

GetGroup (row_idx : long, grp_idx : long) : long
Function returns the group number entered into the matrix into a specified position of the selected row.
Available since version 1.7.

SetGroup (row_idx : long, grp_idx : long, grp_number : long)

Function enters a specified number into the indicated position of the selected row. Only the groups whose nature is compatible with the relation nature can be saved in the relation.
Available since version 1.7.

AddGroup (row_idx : long, grp_number : long)
Function adds a specified number group to the indicated row (at the end of the row). Only the groups whose nature is compatible with the relation nature can be added to the relation.
Available since version 1.7.

SetOperator (row_idx : long, oper_id : IRobotCodeCmbOperator)
Function sets the operator for the selected matrix row.
Available since version 1.7.

GetOperator (row_idx : long) : IRobotCodeCmbOperator
Function returns the operator for the indicated row of relation definition matrix.
Available since version 1.7.

NewRow (oper_id : IRobotCodeCmbOperator) : long
Function creates a new row of the relation definition matrix and returns its index.
Available since version 1.7.

RemoveRow (row_idx : long)
Function deletes a row with the indicated index from the relation definition matrix. The numbers of the remaining rows are updated automatically so that they constitute a sequence of numbers from 1 to RowCount.
Available since version 1.7.

Clear ()
Function deletes all the rows of the relation definition matrix.
Available since version 1.7.

IRobotCodeCmbGroupRelationServer

Server managing relations among groups.

Operations:

Get (rel_nature : IRobotCaseNature) : IRobotCodeCmbGroupRelation
Function returns the definition of the relation with the specified nature.
Available since version 1.7.

Set (rel_def : IRobotCodeCmbGroupRelation)
Function saves a specified group relation definition. The previous group relation definition for the same nature will be replaced.
Available since version 1.7.

IRobotCodeCmbActiveCaseInfo

Properties:

IsSelected : bool
flag indicating if the case will be taken into account while generating code combinations
Available since version 1.7.

CaseNumber : long
number of the load case
Available since version 1.7.

CaseNature : IRobotCaseNature
nature of the load case
Available since version 1.7.

GroupNumber : long
number of the group to which the case belongs
Available since version 1.7.

Coefficient : double
factor
Available since version 1.7.

Snow/wind loads

IRobotSnowWindParams

Set of the non-code-specific parameters describing the manner of generating snow/wind loads.

Properties:

BaySpacing : double
Available since version 1.7.

TotalDepth : double
Available since version 1.7.

Envelope : string
Available since version 1.7.

WithParapets : bool
Available since version 1.7.

WithCavities : bool
Available since version 1.7.

BaseOnGround : bool
Available since version 1.7.

IsolatedRoofs : bool
Available since version 1.7.

IsWind : bool
Available since version 1.7.

IsSnow : bool
Available since version 1.7.

IRobotSnowWindEngine

Generator of snow/wind loads.

Properties:

Params : IRobotSnowWindParams
parameters of snow/wind generation (code independent)
Available since version 1.7.

CodeParams : IDispatch
Available since version 1.7.

Operations:

Generate () : IRobotNumbersCollection
Function generates snow/wind loads on the basis of parameters set earlier.
Function returns the collection of numbers of generated load cases.
Available since version 1.7.

GenerateStruct3D (_gen_params : IRobotSWStruct3DGenParams) : IRobotSWStruct3D
The function generates a 3D frame model based on defined parameters.

Generate3D (_struct_3d : IRobotSWStruct3D) : IRobotNumbersCollection
The function generates snow/wind loads for the given 3D frame model. The function returns a collection of numbers of generated load cases.

IRobotSWCodePLParams

Set of parameters for snow/wind loads for Polish code.

Properties:

StructureHeight : double
structure height
Available since version 1.7.

ReferenceLevel : double
Available since version 1.7.

Altitude : double
Altitude above sea level
Available since version 1.7.

NodalLoadsForAllBars : bool
flag indicating the method of modifying loads into nodal loads; If it is set as a value different from zero (True), then the loads for all the bars in the envelope will be modified.
Available since version 1.7.

NodalLoadsForBarsList : string
selection list describing bars for which loads will be changed to nodal loads
Available since version 1.7.

WindZone : IRobotSWCodePLWindZone
Available since version 1.7.

WindSite : IRobotSWCodePLWindSite
Available since version 1.7.

WindPressureAutomatic : bool
Available since version 1.7.

WindPressure : double
Available since version 1.7.

WindPressureDistribOnHeight : IRobotSWCodePLWindPressDistribType
distribution of wind pressure with respect to height
Available since version 1.7.

WindMultipleRoofs : bool
Available since version 1.7.

WindDynamicAction : bool
Available since version 1.7.

WindDynamicDecrement : double
Available since version 1.7.

WindDynamicPeriod : double
Available since version 1.7.

SnowZone : IRobotSWCodePLSnowZone
Available since version 1.7.

SnowPressureAutomatic : bool
Available since version 1.7.

SnowPressure : double
Available since version 1.7.

SnowRedistribution : bool
Available since version 1.7.

PermLeftSide : double
Available since version 1.7.

PermRightSide : double
Available since version 1.7.

PermFront : double
Available since version 1.7.

PermRear : double
Available since version 1.7.

IsolatedRoofs : IRobotNumbersCollection
table containing numbers of recognized bars
Available since version 1.7.

Operations:

WindBarCoeffGet (bar_num : long) : double
Function returns the coefficient value for a bar with the specified number. Coefficients are determined for the bars in the envelope.
Available since version 1.7.

WindBarCoeffSet (bar_num : long, bar_coeff : double)
Function sets the coefficient value for a bar with the specified number. Coefficients can be determined only for the bars in the envelope.
Available since version 1.7.

SnowBarCoeffGet (bar_num : long) : double
Available since version 1.7.

SnowBarCoeffSet (bar_num : long, bar_coef : double)
Available since version 1.7.

IsolatedRoofSetLocation (bar_num : long, loc : double, _relative : bool)
Available since version 1.7.

IsolatedRoofGetLocation (bar_num : long, relative : bool) : double
Available since version 1.7.

IRobotSWCodePLWindZone

Properties:

I_SWCPLWZ_I : = 0
Available since version 1.7.

I_SWCPLWZ_II : = 1
Available since version 1.7.

I_SWCPLWZ_IIA : = 2
Available since version 1.7.

I_SWCPLWZ_IIB : = 3
Available since version 1.7.

I_SWCPLWZ_III : = 4
Available since version 1.7.

IRobotSWCodePLWindSite

Properties:

I_SWCPLWS_A : = 0
Available since version 1.7.

I_SWCPLWS_B : = 1
Available since version 1.7.

I_SWCPLWS_C : = 2
Available since version 1.7.

IRobotSWCodePLSnowZone

Properties:

I_SWCPLSZ_I : = 0
Available since version 1.7.

I_SWCPLSZ_II : = 1
Available since version 1.7.

I_SWCPLSZ_III : = 2
Available since version 1.7.

I_SWCPLSZ_IV : = 3
Available since version 1.7.

IRobotSWCodeFRParams

Parameters of snow/wind load generation for the French code

Properties:

StructureHeight : double
Available since version 1.7.

ReferenceLevel : double
Available since version 1.7.

Altitude : double
Available since version 1.7.

NodalLoadsForAllBars : bool
Available since version 1.7.

NodalLoadsForBarsList : string
Available since version 1.7.

RiseOfRoofAutomatic : bool
Available since version 1.7.

RiseOfRoof : double
Available since version 1.7.

WindRegion : string
name of wind region
Available since version 1.7.

WindSite : IRobotSWCodeFRWindSite
Available since version 1.7.

WindType : IRobotSWCodeFRWindType
Available since version 1.7.

WindCoastalArea : bool
Available since version 1.7.

WindPressureVariable : bool
Available since version 1.7.

WindPressureAutomatic : bool
Available since version 1.7.

WindPressure : double
Available since version 1.7.

WindDeltaCoeffAutomatic : bool
Available since version 1.7.

WindDeltaCoeff : double
Available since version 1.7.

WindMultipleRoof : bool
Available since version 1.7.

WindFacadeOffset : bool
Available since version 1.7.

WindDynamicAction : bool
Available since version 1.7.

SnowRegion : string
Symbol of snow region
Available since version 1.7.

SnowPressureNormalAutomatic : bool
Available since version 1.7.

SnowPressureNormal : double
Available since version 1.7.

SnowPressureExtremeAutomatic : bool
Available since version 1.7.

SnowPressureExtreme : double
Available since version 1.7.

SnowRedistribution : bool
Available since version 1.7.

SnowGutterBars : string
Available since version 1.7.

PermLeftSide : double
Available since version 1.7.

PermRightSide : double
Available since version 1.7.

PermFront : double
Available since version 1.7.

PermRear : double
Available since version 1.7.

PermDoorLeftSidePresent : bool
Available since version 1.7.

PermDoorLeftSide : double

Available since version 1.7.

PermDoorRightSidePresent : bool
Available since version 1.7.

PermDoorRightSide : double
Available since version 1.7.

PermDoorFrontPresent : bool
Available since version 1.7.

PermDoorFront : double
Available since version 1.7.

PermDoorRearPresent : bool
Available since version 1.7.

PermDoorRear : double
Available since version 1.7.

SurfaceLower : IRobotSWCodeFRSurfaceType
Available since version 1.7.

SurfaceUpper : IRobotSWCodeFRSurfaceType
Available since version 1.7.

OpenStructure : bool
Available since version 1.7.

IsolatedRoofs : IRobotNumbersCollection
Available since version 1.7.

SnowIsWaterOutflow : bool
Available since version 1.7.

SnowWaterOutflow : double
water outflow
Available since version 1.7.

SnowObstacles : string
obstacles - list of bars
Available since version 1.7.

WindDynamicActionPeriod : double
Available since version 1.7.

WindDynamicActionCoeff : double
Available since version 1.7.

WindDynamicActionCoeffAutomatic : bool
Available since version 1.7.

WindDynamicActionSteelStructure : bool
Available since version 1.7.

SnowType : IRobotSWCodeFRSnowType
Available since version 1.7.

WindPressureCeCiMinimum : bool
Available since version 1.7.

Operations:

WindBarCoeffGet (bar_num : long) : double
Available since version 1.7.

WindBarCoeffSet (bar_num : long, _coeff : double)
Available since version 1.7.

SnowBarCoeffGet (bar_num : long) : double
Available since version 1.7.

SnowBarCoeffSet (bar_num : long, _coeff : double)
Available since version 1.7.

IsolatedRoofGetLocation (bar_num : long, _relative : bool) : double
Available since version 1.7.

IsolatedRoofSetLocation (bar_num : long, _location : double, _relative : bool)
Available since version 1.7.

IRobotSWCodeFRWindSite

Properties:

I_SWCFRWS_OBSCURED : = 0
Available since version 1.7.

I_SWCFRWS_NORMAL : = 1
Available since version 1.7.

I_SWCFRWS_EXPOSED : = 2
Available since version 1.7.

IRobotSWCodeFRWindType

Properties:

I_SWCFRWT_NORMAL : = 0
Available since version 1.7.

I_SWCFRWT_EXTREME : = 1
Available since version 1.7.

IRobotSWCodeFRSurfaceType

Properties:

I_SWCFRST_SMOOTH_OR_CORRUGATED : = 0
Available since version 1.7.

I_SWCFRST_FOLDED_OR_CORRUGATED : = 1
Available since version 1.7.

I_SWCFRST_RIBBED : = 2
Available since version 1.7.

IRobotSWCodeFRSnowType

Properties:

I_SWCFRST_NORMAL : = 1
Available since version 1.7.

I_SWCFRST_ACCIDENTAL : = 2
Available since version 1.7.

I_SWCFRST_NORMAL_AND_ACCIDENTAL : = 0
Available since version 1.7.

IRobotSWCodePLWindPressDistribType

Properties:

I_SWCPLWPDT_CONSTANT : = 0
Available since version 1.7.

I_SWCPLWPDT_VARIABLE : = 1
Available since version 1.7.

IRobotSWCodeECParams

Parameters of snow/wind load generation for Eurocode.

Available since version 4.

Properties:

StructureHeight : double

Available since version 4.

ReferenceLevel : double

Available since version 4.

Altitude : double

Available since version 4.

NodalLoadsForAllBars : bool

Available since version 4.

NodalLoadsForBarsList : string

Available since version 4.

LeftWind2NordAngle : double

Available since version 4.

GlobalCDIR : bool

Available since version 4.

StructureAgeCode : string

Available since version 4.

StructureAge : double

Available since version 4.

StructureP : double

Available since version 4.

WindCDIR : double

Available since version 4.

WindCALT : double

Available since version 4.

WindCTEM : double

Available since version 4.

WindPressureAutomatic : bool

Available since version 4.

WindQref : double

Available since version 4.

WindVref0 : double

Available since version 4.

WindQref0 : double

Available since version 4.

WindQref0p : double

Available since version 4.

WindCtAuto : bool

Available since version 4.

WindSiteType : IRobotSWCodeECSiteType

Available since version 4.

WindCt : double

Available since version 4.

GroundType : IRobotSWCodeECGroundType

Available since version 4.

WindKT : double

Available since version 4.

WindZ0 : double

Available since version 4.

WindZMin : double

Available since version 4.

WindE : double

Available since version 4.

WindAutoCd : bool

Available since version 4.

WindCdType : IRobotSWCodeECCdType

Available since version 4.

WindCd : double

Available since version 4.

SnowPressureNormal : double

Available since version 4.

SnowPressureExtreme : double

Available since version 4.

SnowRedistribution : bool

Available since version 4.

SnowGutterBars : string

Available since version 4.

PermLeftSide : double

Available since version 4.

PermRightSide : double

Available since version 4.

PermFront : double

Available since version 4.

PermRear : double

Available since version 4.

PermDoorLeftSidePresent : bool

Available since version 4.

PermDoorLeftSide : double

Available since version 4.

PermDoorRightSidePresent : bool

Available since version 4.

PermDoorRightSide : double

Available since version 4.

PermDoorFrontPresent : bool

Available since version 4.

PermDoorFront : double

Available since version 4.

PermDoorRearPresent : bool

Available since version 4.

PermDoorRear : double

Available since version 4.

Operations:

WindBarCoeffGet (_bar_num : long) : double

Available since version 4.

WindBarCoeffSet (_bar_num : long, _coeff : double)

Available since version 4.

SnowBarCoeffGet (_bar_num : long) : double

Available since version 4.

SnowBarCoeffSet (_bar_num : long, _coeff : double)

Available since version 4.

IRobotSWCodeECSiteType

Available since version 4.

Properties:

I_SWCECST_TYPE_I : = 1

Available since version 4.

I_SWCECST_TYPE_II : = 2

Available since version 4.

I_SWCECST_TYPE_III : = 3

Available since version 4.

I_SWCECST_TYPE_IV : = 4

Available since version 4.

I_SWCECST_TYPE_V : = 5

Available since version 4.

I_SWCECST_TYPE_VI : = 6

Available since version 4.

IRobotSWCodeECGroundType

Available since version 4.

Properties:

I_SWCECGT_TYPE_I : = 1

Available since version 4.

I_SWCECGT_TYPE_II : = 2

Available since version 4.

I_SWCECGT_TYPE_III : = 3

Available since version 4.

I_SWCECGT_TYPE_IV : = 4

Available since version 4.

IRobotSWCodeECCdType

Available since version 4.

Properties:

I_SWCECCT_TYPE_I : = 1

Available since version 4.

I_SWCECCT_TYPE_II : = 2

Available since version 4.

I_SWCECCT_TYPE_III : = 3

Available since version 4.

Generating 3D loads

IRobotSWStruct3D

Structure model defined for generation of 3D snow/wind loads. The model describes a 3D frame generated by copying n times a 2D frame in the third direction. All structure frames consist of the same number of elements. Corresponding elements in individual frames have the same indexes. If an element occurs only in gable frames, then the bar number (Bar) for the corresponding elements of internal frames is ignored.

Properties:

FrameCount : long
number of 2D frames forming a 3D frame

FrameElemCount : long
number of elements of a single frame

Operations:

GetFrame (_frame_idx : long) : IRobotSWStruct3DFrame
The function returns the frame with the specified index. The index of the first (front) frame equals 1, whereas the index of the last (rear) frame equals FrameCount.

SetFrame (_frame_idx : long, _frame : IRobotSWStruct3DFrame)
The function defines the frame with the specified index. Frame indexes are values from the interval [1, FrameCount].

IRobotSWStruct3DElement

Element of a 3D structure model defined for generation of snow/wind loads.

Properties:

Bar : long
bar number

IsFacadeOnly : bool
flag indicating that a bar occurs only in a gable frame (front or rear one)

IsFacadeLoaded : bool
flag forcing load generation in the gable frame for a bar

Purlins : IRobotNumbersArray
table with numbers of bars that define purlins adjoining to this bar

IRobotSWStruct3DFrame

Definition of a 2D frame, a component of a 3D frame, that models a structure for the needs of generation of snow/wind loads. The frame elements are indexed with values from the interval [1, ElemCount].

Properties:

ElemCount : long
number of elements defining a frame

Operations:

SetElem (_elem_idx : long, _elem : IRobotSWStruct3DElement)
The function positions the specified frame element.

GetElem (_elem_idx : long) : IRobotSWStruct3DElement
The function returns the frame element with the specified index.

IRobotSWStruct3DGenParams

Parameters of generation of a 3D frame structure model for the needs of definition of moving loads.

Properties:

FrameCount : long
ultimate number of 2D frames

Spacings : IRobotValuesArray
table including distances (in meters) between successive frames; If only one value (with the index 1) is defined in the table, then all frames will be spaced according to that value.

FacadeOnlyBars : string
selection of bars to be copied only in gable walls

FacadeLoadedBars : string
selection of the gable frame bars for which generation of snow/wind loads will be forced

Offsets : bool
Bars : string
selection of bars forming the initial 2D frame

Operations:

SetPurlins (_bar_num : long, _purlins_def : IRobotSWStruct3DPurlinGenParams)
The function defines the method of generation of longitudinal bars for the indicated frame bar.

GetPurlins (_bar_num : long) : IRobotSWStruct3DPurlinGenParams
The function returns parameters of generation of longitudinal bars for the indicated frame bar.

IRobotSWStruct3DPurlinGenParams

Definition of the method of generation of longitudinal bars (purlins) adjoining to the frame bar.

Properties:

SectionName : string
name of the section that will be assigned to bars-purlins

RelativeLocations : bool
flag indicating that location of successive purlins along the bar has been expressed in values relative to the beginning and length of the bar

Locations : IRobotValuesArray
table defining successive points at which purlins should adjoin to the bar

Objects - complex elements of the structure

IRobotObjObjectServer

Derived from IRobotDataObjectServer
Server of object that are structure components

Properties:

FreeNumber : long
first free user number available for the object
Available since version 1.7.

LinearReleases : IRobotLinearReleaseServer
server of linear releases defined for objects
Available since version 2.5.

Mesh : IRobotObjMesh
component managing the finite element mesh on the level of the entire structure model (for all the objects defined in the structure)
Available since version 3.

AutoRecalcHoles : bool
flag switching on/off automatic detection of openings
Available since version 3.5.

Operations:

Create (num : long) : IRobotObjObject
The function creates and returns a new structure object with the user-defined number

FindWithId (unique_id : long) : long
Function returns a number of the object with the specified unique identifier. If the object is not found, then, zero value is returned (0).
Available since version 1.7.

CreateCube (_number : long, _points : IRobotPointsArray, _has_base : bool, _has_top : bool, _divisions : long, _is_volume : bool = true) : IRobotObjObject
Function creates a cube based on the indicated four points - first three points (P1, P2, P3) define the cube base and the fourth point (PH) defines the height. The cube will be created in such a manner so that the PH point is connected by an edge with the P3 point.
Available since version 3.

CreateCylinder (_number : long, _points : IRobotPointsArray, _has_base : bool, _has_top : bool, _divisions : long, _sides : long, _is_volume : bool = true) : IRobotObjObject
Function creates a cylinder based on the indicated four points. First three points define the cylinder base and the fourth point (PH) defines its height. The cylinder will be created in such a manner so that the fourth point (PH) is a center of the second base of the cylinder.

Available since version 3.

CreateCone (_number : long, _points : IRobotPointsArray, _has_base : bool, _divisions : long, _sides : long, _is_volume : bool = true, _has_top : bool = true) : IRobotObjObject

Function creates a cone based on the indicated four points. First three points define the cone base and the fourth point defines position of the vertex.
To create a truncated cone the top radius should be provided as a fifth defining point
Available since version 3.

CreateOnFiniteElems (_finite_elems : string, _first_number : long) : IRobotNumbersCollection

Function creates one or more objects in such a manner so that they cover the indicated list of finite elements. Function returns the collection of the created object numbers.
Available since version 3.

CreatePolyline (_number : long, _points : IRobotPointsArray)

Function creates a polyline of the specified user-defined number based on the indicated points.
Available since version 3.

CreateContour (_number : long, _points : IRobotPointsArray)

Function creates a contour based on the specified points.
Available since version 3.

CreateArc (_number : long, _points : IRobotPointsArray, _creation_type : IRobotGeoArcDefinitionMethod = I_GADM_CENTER_BEGIN_END)

Function creates an arc based on the indicated points.
Available since version 3.

CreateCircle (_number : long, _points : IRobotPointsArray)

Function creates a circle of the specified user-defined number based on three points.
Available since version 3.

BeginMultiOperation ()

Function enables speed-up of generation or modification of a whole object group through appropriate optimization of some operations. After completing object generation, EndMultiOperation() function should be called up to update definitions of objects in a structure.
Available since version 3.5.

EndMultiOperation ()

Function updates definitions of objects in a structure. It should be called up only if BeginMultiOperation() function has been called up earlier.
Available since version 3.5.

CalcArea (_obj_num : long, _part_idx : long = 0) : double

Function calculates area of the indicated object component. If the object component index equals 0, function returns area of the whole object.
Attention:
Before calling the function, mesh of surface finite elements should be generated.
Available since version 4.5.

CalcVol (_obj_num : long, _part_idx : long = 0) : double
Function calculates volume of the indicated solid or its component. If the object component index equals 0, function returns volume of the whole solid.
Attention:
Before calling the function, mesh of volumetric finite elements should be generated.
Available since version 4.5.

IsVolume (_obj_num : long) : bool
Function returns the True value, if the indicated object is a solid.
Available since version 4.5.

GetName (_obj_num : long) : string
Function returns a name of the object of the given number.
Available since version 7.5.

GetNameTemplate (_obj_num : long) : string
Function returns the name pattern of an object of the specified number.
Available since version 7.5.

SetNameTemplate (_obj_num : long, _name_tmpl : string)
Function sets a name pattern for the specified object.
Available since version 7.5.

IRobotObjObject

Derived from IRobotDataObject

Properties:

Main : IRobotObjPartMain
main object component

PartsCount : long
number of object components

UniqueId : long
unique object identifier
Available since version 1.7.

Reference : IRobotObjPartReference
main component reference
Available since version 2.0.

FiniteElems : string
list of all finite elements assigned to the object
Available since version 2.5.

Mesh : IRobotObjMesh
component managing the finite element mesh for this object

Available since version 3.

Nodes : string
list of nodes belonging to the object
Available since version 4.

IsVolume : bool
flag (read-only) indicating if the object is a solid
Available since version 4.5.

Name : string
object name
Available since version 7.5.

NameTemplate : string
object name pattern
Available since version 7.5.

Operations:

GetPartType (part_num : long) : IRobotObjPartType
The function returns the type of the indicated object part

GetPart (_part_num : long) : IRobotObjPart
Function returns the indicated object component. Components are indexed from 1 to their number.

Initialize ()
The function initializes or rebuilds the object on the basis of the base geometry and transformations defined in the main component.

Update ()
The function updates definition of the object once its attributes are changed.

CalcArea () : double
Function calculates the object area.
Available since version 4.5.

CalcVol () : double
Function calculates the solid volume.
Available since version 4.5.

IRobotObjPartType

Properties:

I_OPT_MAIN : = 0
the main part of the object

I_OPT_REFERENCE : = 1
reference - the image of the main part of the object after transformation

I_OPT_SIDE : = 2
side - side object component

IRobotObjPart

Interface describing an object component.

Properties:

Attribs : IRobotObjAttributes
attributes of the object component

Type : IRobotObjPartType
object component type
Available since version 2.0.

Operations:

GetGeometry () : IRobotGeoObject
Function returns defintion of the object component geometry.
Available since version 2.0.

IRobotObjPart2

Derived from IRobotObjPart
Extended definition of the object component.

Available since version 2.5.

Properties:

FiniteElems : string
text containing the list of finite elements assigned to this object component

Available since version 2.5.

Nodes : string

Available since version 4.

Operations:

CalcArea () : double
Function calculates area of the object component.
Available since version 4.5.

CalcVol () : double
Function calculates volume of the solid component.
Available since version 4.5.

IRobotObjPartMain

Derived from IRobotObjPart
Main object component. It contains definition of the base geometry and geometric transformations which result in the geometry of the entire object. All object edges are also saved in the main component.

Properties:

Geometry : IRobotGeoObject
definition of the base object geometry
Available since version 2.0.

Modifications : IRobotObjModificationCollection
collection of transformations of the base object geometry
Available since version 2.0.

CurveDiv : IRobotGeoCurveDiv
definition of object discretization
Available since version 2.0.

DefPoints : IRobotGeoPoint3DCollection
collection of points defining the base object geometry
Available since version 2.0.

Edges : IRobotObjEdgeCollection
collection of all object edges
Available since version 2.0.

FiniteElems : string
list of finite elements assigned to the main object component
Available since version 2.5.

ModelPoints : IRobotGeoPoint3DCollection
Characteristic points used for contour discretization
Available since version 3.

Nodes : string

Available since version 4.

Operations:

ClearModifications ()

Function deletes all transformations from the collection of transformations.
Available since version 2.0.

AddModification (_mod_def : IRobotObjModification)
Function adds the indicated transformation to the collection.
Available since version 2.0.

CalcArea () : double
Function calculates area of the object component.
Available since version 4.5.

CalcVol () : double
Function calculates volume of the solid component.
Available since version 4.5.

IRobotObjPartReference

Derived from IRobotObjPart
Object component definition which is the reference of the main component generated due to transformations.

Properties:

FiniteElems : string
list of finite elements assigned to this object component
Available since version 2.5.

Nodes : string

Available since version 4.

Operations:

CalcArea () : double
Function calculates area of object component.
Available since version 4.5.

CalcVol () : double
Function calculates volume of solid component.
Available since version 4.5.

IRobotObjEdge

interface describing an object edge

Properties:

Path : IRobotGeoPoint3DCollection

edge geometry

Operations:

HasLabel (lab_type : IRobotLabelType) : bool

SetLabel (lab_type : IRobotLabelType, lab_name : string)

GetLabel (lab_type : IRobotLabelType) : IRobotLabel

GetLabelName (lab_type : IRobotLabelType) : string

RemoveLabel (lab_type : IRobotLabelType)

GetLabels () : IRobotCollection

DivideN (_n : long, _only_gen_nodes : bool = false)
Function divides the edge into N equal parts.
Available since version 4.5.

Divide (_x : double, _relative : bool = false, _only_gen_node : bool = false)
Fuction divides the edge at the indicated point. If the _relative flag is set as True, then the _x parameter determines a relative distance between the division point and edge beginning - a value contained in the interval (0, 1). Otherwise, _x determines a distance between the division point and edge beginning expressed in meters.
Available since version 4.5.

DivideByPlane (_val : double, _axis : IRobotGeoCoordinateAxis, _only_gen_node : bool = false)
Function divides the edge by cutting it with the indicated plane of the defined coordinate on the normal axis.
Available since version 4.5.

IRobotObjAttributes

The interface describing additional attributes of an object or its part

Properties:

Meshed : bool
flag determining if mesh is to be generated on an object component

DirZ : bool
the flag allowing change of the object component orientation to the opposite (change of the local Z axis sense)

Operations:

SetLabel (lab_type : IRobotLabelType, lab_name : string)

HasLabel (lab_type : IRobotLabelType) : bool

GetLabelName (lab_type : IRobotLabelType) : string

RemoveLabel (lab_type : IRobotLabelType)

GetLabels () : IRobotCollection

SetDirX (def_type : IRobotObjLocalXDirDefinitionType, _x : double, _y : double, _z : double)
Function defines the local X axis direction in the defined coordinate system. Additional function parameters determine the coordinates of the X axis direction vector for definition in the Cartesian system or the coordinates of the system beginning for definition in the polar system.
Available since version 1.7.

GetDirX (_x : double*, _y : double*, _z : double*) : IRobotObjLocalXDirDefinitionType
Function reads the local X axis direction. Read values are the coordinates of the X axis direction vector for definition in the Cartesian system or the coordinates of the system beginning for definition in the polar system. Additionally, the definition method is returned.
Available since version 1.7.

IRobotObjModificationType

Set of identifiers has been defined for modifications which may be performed on specified objects - structure components.

Properties:

I_OMT_NONE : = 17
undefined

I_OMT_EXTRUSION : = 7
extrude

I_OMT_LATHE : = 8
revolve

I_OMT_PYRAMID : = 9
extrude along polyline

IRobotObjOperationType

Set of identifiers has been defined for operations which may be performed on specified objects - structure components.

Properties:

I_OOT_NONE : = 16
undefined

I_OOT_MESH : = 1
deformation

I_OOT_TRANSLATE : = 2
translation

I_OOT_SCALE : = 3
scaling

I_OOT_ROTATE : = 18
rotation

IRobotObjModification

The basic interface describing modification of the initial object geometry

Properties:

Type : IRobotObjModificationType
modification type

Operations : IRobotObjOperationCollection
collection of operations describing transformation of the modification result (so called reference or image)

Filled : bool
the flag that indicates if the modification is filled with volumetric elements

NDiv : long
number of divisions of lateral walls

Operations:

Clear ()
The function clears the table of operations.

Add (operation : IRobotObjOperation)
The function adds a new operation.

IRobotObjOperation

The basic interface describing transformation (translation, rotation, scaling, deformation) of the result of modification (the so called reference or image)

Properties:

Type : IRobotObjOperationType
operation type

IRobotObjOperationCollection

Derived from IRobotCollection
Collection of operations

IRobotObjModifExtrusion

Derived from IRobotObjModification
The interface describing a modification of initial object geometry - extrude.

Properties:

Vector : IRobotGeoPoint3D
direction vector for extrusion

IRobotObjModifLathe

Derived from IRobotObjModification
Interface describing modification of initial geometry of the object - revolve operation

Properties:

AxsP1 : IRobotGeoPoint3D
the first point of revolution axis

AxsP2 : IRobotGeoPoint3D
the second point of revolution axis

Angle : double
revolution angle

IRobotObjModifPyramid

Derived from IRobotObjModification
Interface describing modification of initial object geometry - extrude along polyline

Properties:

Points : IRobotGeoPoint3DCollection
polyline points

Factors : IRobotGeoPoint3DCollection
direction (XYZ) scaling factors

Operations:

AddPoint (point : IRobotGeoPoint3D, factor : IRobotGeoPoint3D)
Function adds a new point of the polyline with a scaling factor

ClearPoints ()
Available since version 1.7.

IRobotObjOperTranslation

Derived from IRobotObjOperation
The interface describing the operation of translation that transforms the result of modification of the initial object geometry.

Properties:

Vector : IRobotGeoPoint3D
translation vector

IRobotObjOperScaling

Derived from IRobotObjOperation
The interface describing the operation of scaling, transforming the result of modification of object geometry

Properties:

Factor : IRobotGeoPoint3D
scaling coefficient in the three main directions

Center : IRobotGeoPoint3D
scaling center

IRobotObjOperRotation

Derived from IRobotObjOperation
Interface describing the operation of rotation that transforms the result of modification of object geometry.

Properties:

AxsP1 : IRobotGeoPoint3D
the first point of rotation axis

AxsP2 : IRobotGeoPoint3D
the second point of rotation axis

Angle : double

rotation angle

IRobotObjOperMeshing

Derived from IRobotObjOperation
The interface describing a deformation that transforms the result of modification of initial object geometry

Properties:

Vectors : IRobotGeoPoint3DCollection
collection of displacement vectors

Points : IRobotNumbersCollection
collection of indexes (numbers) of points that undergo displacement

Operations:

Clear ()
The function clears the collection of vectors and point numbers

Add (point : long, vector : IRobotGeoPoint3D)
The function adds a displacement vector for an indicated point that defines initial object geometry. The initial geometry of each object is defined through a set of points.

IRobotObjModificationCollection

Derived from IRobotCollection
Collection of modifications

IRobotObjEdgeCollection

Derived from IRobotCollection
Collection of edges.

IRobotObjLocalXDirDefinitionType

Type of direction definition for the local X axis of the plate.

Properties:

I_OLXDDT_UNDEFINED : = 0
local X axis direction is not defined (not significant)
Available since version 1.7.

I_OLXDDT_CARTESIAN : = 1
definition of local X axis direction in the Cartesian coordinate system

Available since version 1.7.

I_OLXDDT_POLAR : = 2
definition of local X axis direction in the polar coordinate system
Available since version 1.7.

IRobotObjMesh

Interface allowing management of the finite element mesh.
Available since version 3.

Properties:

IsGenerated : bool
flag indicating if the finite element mesh has been generated
Available since version 3.

Freeze : bool
flag enabling mesh freezing/unfreezing
Available since version 3.

Params : IRobotMeshParams
Parameters of mesh generation
Available since version 3.

Operations:

Generate ()
Function generates a finite element mesh.
Available since version 3.

Remove ()
Function removes a finite element mesh.
Available since version 3.

GetQuality (_out_q1 : double*, _out_q2 : double*, _out_fes : string*, _in_precision : double = 0.5, _in_objs : string = "")
Function checks a quality of a finite element mesh.
Available since version 3.

SetBasePoints (_model_points : IRobotNumbersArray)
Function sets base points for a finite element mesh.
Available since version 3.

GetBasePoints () : IRobotNumbersArray
Function returns base points for a finite element mesh.
Available since version 3.

Complex attributes of an object

IRobotThicknessData

Thickness definition

Properties:

ThicknessType : IRobotThicknessType
Available since version 1.7.

ElasticFoundation : double
Available since version 1.7.

Uplift : IRobotThicknessUpliftType
Available since version 1.7.

Data : Object
set of data that are specific for the selected thickness type;
parameters describing thickness of the I_TT_HOMOGENEOUS type are defined by means of RobotThicknessHomoData, whereas for the I_TT_ORTHOTROPIC type RobotThicknessOrthoData is defined
Available since version 1.7.

MaterialName : string
name of the material connected with thickness
Available since version 2.0.

IRobotThicknessType

Properties:

I_TT_HOMOGENEOUS : = 0
Available since version 1.7.

I_TT_ORTHOTROPIC : = 1
Available since version 1.7.

IRobotThicknessUpliftType

Properties:

I_TUT_NONE : = 0
Available since version 1.7.

I_TUT_MINUS : = 1

Available since version 1.7.

I_TUT_PLUS : = 2
Available since version 1.7.

IRobotThicknessHomoData

Definition of thickness parameters of the I_TT_HOMOGENEOUS type.

Properties:

Type : IRobotThicknessHomoType
Available since version 1.7.

Thick1 : double
thickness value in the P1 point (set only for variable thickness)
Available since version 1.7.

Thick2 : double
thickness value in the P2 point (set only for variable thickness)
Available since version 1.7.

Thick3 : double
thickness value in the P3 point (set only for variable thickness)
Available since version 1.7.

ThickConst : double
thickness value (if the thickness is uniform for the whole slab)
Available since version 1.7.

Operations:

SetP1 (_x : double, _y : double, _z : double)
Available since version 1.7.

GetP1 (_x : double*, _y : double*, _z : double*)
Available since version 1.7.

SetP2 (_x : double, _y : double, _z : double)
Available since version 1.7.

GetP2 (_x : double*, _y : double*, _z : double*)
Available since version 1.7.

SetP3 (_x : double, _y : double, _z : double)
Available since version 1.7.

GetP3 (_x : double*, _y : double*, _z : double*)
Available since version 1.7.

SetReduction (_set : bool, _Ig_cf : double)
Function sets a coefficient of reduction of the moment of inertia.
Available since version 8.

GetReduction (_Ig_cf : double*) : bool
Function returns True value if a reduction coefficient of the moment of inertia has been set. The coefficient value will be entered to the specified parameter.
Available since version 8.

IRobotThicknessHomoType

Properties:

I_THT_CONSTANT : = 0
Available since version 1.7.

I_THT_VARIABLE_ALONG_LINE : = 1
Available since version 1.7.

I_THT_VARIABLE_ON_PLANE : = 2
Available since version 1.7.

IRobotThicknessOrthoType

Properties:

I_TOT_ONE_SIDED_UNIDIR_RIBS : = 1
Available since version 1.7.

I_TOT_DOUBLE_SIDED_UNIDIR_RIBS : = 2
Available since version 1.7.

I_TOT_ONE_SIDED_BIDIR_RIBS : = 3
Available since version 1.7.

I_TOT_UNIDIR_BOX_FLOOR : = 4
Available since version 1.7.

I_TOT_BIDIR_BOX_FLOOR : = 5
Available since version 1.7.

I_TOT_GRILLAGE : = 6
Available since version 1.7.

I_TOT_SLAB_ON_TRAPEZOID_PLATE : = 7
Available since version 1.7.

I_TOT_USER : = 8
geometrical thickness parameters should be described by means of the appropriate matrices
Available since version 1.7.

I_TOT_MATERIAL : = 9
material orthotropy
Available since version 1.7.

IRobotThicknessOrthoDirType

Properties:

I_TODT_DIR_X : = 0
Available since version 1.7.

I_TODT_DIR_Y : = 1
Available since version 1.7.

I_TODT_DIR_Z : = 2
Available since version 1.7.

I_TODT_VECTOR : = 3
Available since version 1.7.

IRobotThicknessOrthoData

Properties:

Type : IRobotThicknessOrthoType
Available since version 1.7.

DirType : IRobotThicknessOrthoDirType
Available since version 1.7.

H : double
Available since version 1.7.

HA : double
Available since version 1.7.

HB : double
Available since version 1.7.

H1 : double
Available since version 1.7.

H2 : double
Available since version 1.7.

A : double
Available since version 1.7.

A1 : double
Available since version 1.7.

A2 : double
Available since version 1.7.

B : double
Available since version 1.7.

B1 : double
Available since version 1.7.

Thick1 : double
Available since version 1.7.

Thick2 : double
Available since version 1.7.

Thick3 : double
Available since version 1.7.

Matrix : IRobotThicknessMatrix
object providing access to stiffness matrix values
Available since version 1.7.

N1 : double
first stiffness coefficient for material orthotropy
Available since version 1.7.

N2 : double
second stiffness coefficient for material orthotropy
Available since version 1.7.

Operations:

SetVector (_x : double, _y : double, _z : double)
Available since version 1.7.

GetVector (_x : double*, _y : double*, _z : double*)
Available since version 1.7.

IRobotThicknessMatrix

Object provides access to stiffness matrix values for orthotropic thicknesses.

Operations:

GetValue (val_id : IRobotThicknessMatrixValue) : double
Function returns a value of the indicated matrix element.
Available since version 1.7.

SetValue (val_id : IRobotThicknessMatrixValue, val : double)
Function sets the value of the indicated matrix element.
Available since version 1.7.

IRobotThicknessMatrixValue

Physical interpretation of individual matrix values is described in the file of the Robot program help.

Properties:

I_TMV_D_XXXX : = 0
defining this quantity makes sense only in the case of shell structures (it is ignored for plates)
Available since version 1.7.

I_TMV_D_XXYY : = 1
defining this quantity makes sense only in the case of shell structures (it is ignored for plates)
Available since version 1.7.

I_TMV_D_YYYY : = 2
defining this quantity makes sense only in the case of shell structures (it is ignored for plates)
Available since version 1.7.

I_TMV_D_XYXY : = 3
defining this quantity makes sense only in the case of shell structures (it is ignored for plates)
Available since version 1.7.

I_TMV_K_XXXX : = 4
Available since version 1.7.

I_TMV_K_XXYY : = 5
Available since version 1.7.

I_TMV_K_YYYY : = 6
Available since version 1.7.

I_TMV_K_XYXY : = 7
Available since version 1.7.

I_TMV_H_XX : = 8
Available since version 1.7.

I_TMV_H_YY : = 9
Available since version 1.7.

IRobotSolidPropertiesData

Solid properties
Available since version 3.

Properties:

MaterialModel : IRobotMaterialModel
Material model
Available since version 3.

E : double
Young's modulus
Available since version 3.

NU : double
Poisson ratio
Available since version 3.

RO : double
Density
Available since version 3.

LX : double
Thermal expansion coefficient
Available since version 3.

DampCoef : double
Damping coefficient
Available since version 3.

Operations:

LoadFromDBase (_material_name : string) : bool
Function reads data characteristic of a material of the indicated name from the material database. If the material of the indicated name is not found in the current material database, then zero value is returned (False).
Available since version 3.

Linear releases

Available since version 2.5.

IRobotLinearReleaseData

Linear release between panels.
Available since version 2.5.

Properties:

UX : IRobotLinearReleaseDefinitionType
definition of the release (UX direction)
Available since version 2.5.

UY : IRobotLinearReleaseDefinitionType
definition of the release (UY direction)
Available since version 2.5.

UZ : IRobotLinearReleaseDefinitionType
definition of the release (UZ direction)
Available since version 2.5.

RX : IRobotLinearReleaseDefinitionType
definition of the release (RX direction)
Available since version 2.5.

KX : double
KX elastic coefficient for the elastic release in the UX direction
Available since version 2.5.

KY : double
KY elastic coefficient for the elastic release in the UY direction
Available since version 2.5.

KZ : double
KZ elastic coefficient for the elastic release in the UZ direction
Available since version 2.5.

HX : double
HX elastic coefficient for the elastic release in the RX direction
Available since version 2.5.

IRobotLinearReleaseDefinitionType

Available methods of release definition for one direction.
Available since version 2.5.

Properties:

I_LRDT_NONE : = 0
no release for the selected direction

Available since version 2.5.

I_LRDT_RELEASED : = 1
release definition for the selected direction
Available since version 2.5.

I_LRDT_MINUS : = 2
release definition in the direction opposite to the local system axis on the panel edge
Available since version 2.5.

I_LRDT_PLUS : = 3
release definition in the direction compatible with the local system axis on the panel edge

Available since version 2.5.

IRobotLinearReleaseServer

Server allowing definition of linear releases between panels. Moreover, it provides access to the list of objects describing all the releases defined.
Available since version 2.5.

Properties:

Count : long
number of defined releases
Available since version 2.5.

Operations:

Set (_edge_obj_num : long, _edge_idx : long, _obj_num : long, _part_idx : long, _label_name : string) : bool
Function defines a linear release for the selected object component on the indicated edge. The edge may belong to object other than the released component. Function returns value different from zero (True) if the release is defined.
Available since version 2.5.

Remove (_def_idx : long)
Function deletes the linear release whose method of definition describes the object with the indicated index on the list of all the releases defined.
Available since version 2.5.

Find (_edge_obj_num : long, _edge_idx : long, _obj_num : long, _part_idx : long) : long
Function returns index of the definition describing the release between the indicated objects. If no release has been defined between the indicated objects, then zero value is returned.
Available since version 2.5.

GetLabel (_def_idx : long) : IRobotLabel
Function returns label describing parameters of the indicated linear release.

Available since version 2.5.

Get (_def_idx : long) : IRobotLinearReleaseDef
Function returns the objects containing parameters of the linear release definition. By activating this function with the successive indexes from 1 to Count, the user may obtain information about all the defined linear releases between panels.
Available since version 2.5.

FindObject (_obj_num : long, _part_idx : long, _start_def_idx : long = 1) : long
Function returns index of the first located linear release definition for the specified object component. The list of the defined linear releases is searched beginning with the indicated start index. If the linear release is not located for the indicated object component, then zero value is returned.
Available since version 2.5.

FindEdge (_edge_obj : long, _edge_idx : long, _start_def_idx : long = 1) : long
Function returns index of the first located linear release definition on the indicated edge. The list of the defined linear releases is searched beginning with the indicated start index. If the release is not located on the indicated edge, then zero value is returned.
Available since version 2.5.

FindLabel (_label_name : string, _start_def_idx : long = 1) : long
Function returns index of the first located definition of the linear release with the indicated name. The list of the defined linear releases is searched beginning with the indicated start index. If the release is not located, then zero value is returned.

Available since version 2.5.

SetMany (_deflist : IRobotLinearReleaseDefList)
Function defines linear releases according to the specified list.
Available since version 8.

IRobotLinearReleaseDef

Object describing definition parameters of the linear release between panels.
The release is defined for the selected object component on the indicated edge. The edge may belong to object other than the released component.
Available since version 2.5.

Properties:

EdgeObject : long
number of the object including the edge along which the release is defined
Available since version 2.5.

EdgeIdx : long
index of the edge in the object
Available since version 2.5.

Object : long
number of the object for the component of which the release is defined
Available since version 2.5.

PartIdx : long
index of the object component for which the release is defined
Available since version 2.5.

LabelName : string
name of the label storing release parameters
Available since version 2.5.

IRobotLinearReleaseDefList

List of definitions of linear releases.
Available since version 8.

Properties:

Count : long

Available since version 8.

Operations:

Add (_edge_obj_num : long, _edge_idx : long, _obj_num : long, _part_idx : long, _label_name : string) : long
Function adds a linear release definition to the list. Function returns an index of the definition added to the list.
Available since version 8.

Clear ()
Function deletes all elements from the list.
Available since version 8.

Get (_idx : long, _edge_obj_num : long*, _edge_idx : long*, _obj_num : long*, _part_idx : long*, _label_name : string*) : bool
Function makes available a linear release definition of the specified index.
Available since version 8.

Remove (_idx : long)
Function deletes a linear release definition of the specified index from the list.
Available since version 8.

Available since version 7.5.

IRobotGroupObjectServer

Servers allows operations of grouping / ungrouping objects.
Available since version 7.5.

Operations:

GroupSelected () : long
Function groups (combines into one object) structure elements belonging to the current selection. Function returns the number of a created object.
Available since version 7.5.

GroupGiven (_object_sel : string, _fe_sel : string, _node_sel : string, _auto_include : bool = True) : long
Function groups (combines into one object) structure elements described by selection lists. Function returns the number of a created object.
Available since version 7.5.

FindFirst () : long
Function returns the number of the first object grouping structure elements. If no objects of this type have been defined, function returns zero value.
Available since version 7.5.

FindNext (_prev_group_obj : long) : long
Function returns the number of the next object grouping structure elements. If no more grouping objects have been defined, function returns zero value.
Available since version 7.5.

Explode (_group_obj : long)
Function ungroups a group of structure elements of the specified number.
Available since version 7.5.

GetContents (_group_obj : long, _object_sel : string*, _fe_sel : string*, _node_sel : string*)
Function returns selection lists of structure elements included in the specified "grouped" object.
Available since version 7.5.

Finite elements

Available since version 2.5.

IRobotFiniteElement

Derived from IRobotDataObject
Interface describing planar finite element.
Available since version 2.5.

Properties:

UniqueId : long
unique element identifier
Available since version 2.5.

FeType : IRobotFiniteElementType
finite element type
Available since version 2.5.

Nodes : IRobotFiniteElementNodes
nodes defining a finite element
Available since version 2.5.

ObjectNumber : long
number of the object including the finite element
Available since version 2.5.

ObjectPartIdx : long
index of the object component including the element
Available since version 2.5.

Operations:

CalcArea () : double
Function calculates a finite element area.
Available since version 7.5.

IRobotFiniteElementType

Available types of finite elements.
Available since version 2.5.

Properties:

I_FET_T3 : = 3
three-node element
Available since version 2.5.

I_FET_Q4 : = 4
four-node element
Available since version 2.5.

I_FET_T6 : = 1
six-node element
Available since version 2.5.

I_FET_Q8 : = 2
eight-node element
Available since version 2.5.

I_FET_VOL_B8 : = 5

Available since version 7.5.

I_FET_VOL_T4 : = 6

Available since version 7.5.

I_FET_VOL_W6 : = 7

Available since version 7.5.

I_FET_VOL_B20 : = 8

Available since version 7.5.

I_FET_VOL_T10 : = 9

Available since version 7.5.

I_FET_VOL_W15 : = 10

Available since version 7.5.

IRobotFiniteElementNodes

Collection of nodes defining finite element.
Available since version 2.5.

Properties:

Count : long
number of nodes in the collection (simultaneously, it is the index of the last node)
Available since version 2.5.

Operations:

Get (_node_idx : long) : long
Function returns number of the node with the specified index in the collection.
Available since version 2.5.

Set (_node_idx : long, _node_num : long) : bool
Function inserts the specified node onto the indicated index in the collection of nodes defining finite element.
Available since version 2.5.

SetAll (_nodes_array : IRobotNumbersArray)
Function defines all nodes of the finite element.
Available since version 2.5.

GetAll () : IRobotNumbersArray
Function returns the table with numbers of nodes defining the finite element.
Available since version 2.5.

IRobotFiniteElementServer

Derived from IRobotDataObjectServer
Server of finite elements.
Available since version 2.5.

Properties:

FreeNumber : long
first avaialble user-defined number that can be assigned to a finite element
Available since version 2.5.

Operations:

MeshConsolidate (_coeff : double, _sel : IRobotSelection, _auto_freeze : bool = true)
Function performs mesh consolidation for the indicated finite element list.
Available since version 3.

MeshConcentrate (_type : IRobotMeshRefinementType, _sel : IRobotSelection, _auto_freeze : bool = true)
Function carries out mesh refinement for the indicated selection of finite elements.
Available since version 3.

Create (_number : long, _nodes : IRobotNumbersArray, _update_model : bool = True)
Function creates a new finite element defined on the indicated nodes.
Available since version 3.

Update ()
Function reconstructs relations between finite elements and structure nodes.
It should be activated, if the Create function with delayed model update, i.e., with parameter_ update_model set as False, has been used earlier.
Available since version 3.

CalcArea (_ele_num : long) : double
Function calculates an area of the finite element of the given number
Available since version 7.5.

Mesh generator

Available since version 3.

IRobotMeshType

Types of generated finite element mesh (surface or volume).
Available since version 3.

Properties:

I_MT_FINE : = 1
fine meshing type
Available since version 3.

I_MT_COARSE : = 2
coarse meshing type
Available since version 3.

I_MT_NORMAL : = 3
normal meshing type
Available since version 3.

I_MT_USER : = 4
user-defined meshing type
Available since version 3.

IRobotMeshForcingRatio

Forcing ratio of selected meshing parameter
Available since version 3.

Properties:

I_MFR_NONE : = 1
forcing ratio - none
Available since version 3.

I_MFR_ANY : = 2
any forcing ratio
Available since version 3.

I_MFR_PROPOSED : = 3
proposed forcing ratio
Available since version 3.

I_MFR_RECOMMENDED : = 4
recommended forcing ratio
Available since version 3.

I_MFR_FORCED : = 5
forced forcing ratio
Available since version 3.

IRobotMeshImplementDegree

Degrees of the selected method application while generating finite element mesh.
Available since version 3.

Properties:

I_MIR_OFTEN : = 1

Available since version 3.

I_MIR_RARELY : = 2

Available since version 3.

I_MIR_NEVER : = 3

Available since version 3.

IRobotMeshPanelDivType

Methods of contour division for the Coons method of finite element mesh generation.
Available since version 3.

Properties:

I_MPDT_TRIANG_IN_TRIANG : = 1
division allowing generation of triangular finite elements in a selected triangular contour
Available since version 3.

I_MPDT_TRIANG_AND_SQUARE_IN_TRIANG : = 2
division allowing generation of triangular and square finite elements in a selected triangular contour

Available since version 3.

I_MPDT_TRIANG_AND_TRAPEZ_IN_TRIANG : = 3
division allowing generation of triangular and rhombic finite elements in a selected triangular contour

Available since version 3.

I_MPDT_SQUARE_IN_RECT : = 4
division allowing generation of square finite elements in a selected rectangular contour

Available since version 3.

I_MPDT_TRIANG_IN_RECT : = 5
division allowing generation of triangular finite elements in a selected rectangular contour

Available since version 3.

IRobotMeshSurfaceFEType

Finite element types applied while generating a surface element mesh.
Available since version 3.

Properties:

I_MSFET_3NODE_TRIANG : = 1
3-node triangular surface finite elements
Available since version 3.

I_MSFET_4NODE_QUADRIL : = 2
4-node quadrilateral surface finite elements
Available since version 3.

I_MSFET_6NODE_TRIANG : = 3
6-node triangular surface finite elements
Available since version 3.

I_MSFET_8NODE_QUADRIL : = 4
8-node quadrilateral surface finite elements
Available since version 3.

IRobotMeshVolumetricFEType

Finite element types applied during generation of volumetric element mesh.
Available since version 3.

Properties:

I_MVFET_4NODE_TETRAHED : = 1
4-node tetrahedral volumetric finite elements
Available since version 3.

I_MVFET_8NODE_HEXAHED : = 2
8-node hexahedral volumetric finite elements
Available since version 3.

IRobotMeshDelaunayType

Available variants of Delaunay's method.
Available since version 3.

Properties:

I_MDT_DELAUNAY : = 1
standard Delaunay's method
Available since version 3.

I_MDT_KANG : = 2
Kang's method; selection of this variant indicates that the finite element mesh will be generated only on a contour near emitters according to the adopted method parameters (H0, Hmax, and Q)
Available since version 3.

I_MDT_DELAUNAY_AND_KANG : = 3
Delaunay and Kang method; selection of this variant indicates that near emitters the finite element mesh will be generated by means of Kang's method, whereas on the remaining contour - according to Delaunay's method
Available since version 3.

IRobotMeshMethodType

Acceptable meshing methods.
Available since version 3.

Properties:

I_MMT_DELAUNAY : = 1
Delaunay's method
Available since version 3.

I_MMT_COONS : = 2
Coons method
Available since version 3.

IRobotMeshGenerationType

Methods of finite element mesh generation.
Available since version 3.

Properties:

I_MGT_AUTOMATIC : = 1
Automatic generation
Available since version 3.

I_MGT_USER : = 2
generation with user-defined parameters applied
Available since version 3.

I_MGT_ELEMENT_SIZE : = 3

Available since version 4.5.

IRobotMeshMethod

A set of parameters defining a mesh generation method and associated quantities.
Available since version 3.

Properties:

Method : IRobotMeshMethodType
Meshing method
Available since version 3.

ForcingRatio : IRobotMeshForcingRatio
forcing ratio for the selected method
Available since version 3.

ImplementDegree : IRobotMeshImplementDegree
degree of application for the selected method during mesh generation
Available since version 3.

IRobotMeshGeneration

Parameters of mesh generation
Available since version 3.

Properties:

Type : IRobotMeshGenerationType
Meshing type
Available since version 3.

Division1 : double
division parameter determining the number of elements to be generated on the first contour edge (between first and second contour vertex)
Available since version 3.

Division2 : double
division parameter determining the number of elements to be generated on the second contour edge (between second and third contour vertex)

Available since version 3.

ElementSize : double
expected length of the finite element edge (in meters)
Available since version 4.5.

IRobotMeshCoonsParams

Parameters of Coons method.
Available since version 3.

Properties:

PanelDivisionType : IRobotMeshPanelDivType
manner of contour division for Coons method
Available since version 3.

ForcingRatio : IRobotMeshForcingRatio
forcing ratio for the selected method of contour division
Available since version 3.

IRobotMeshSurfaceFiniteElems

Definition of surface finite elements applied during mesh generation.
Available since version 3.

Properties:

Type : IRobotMeshSurfaceFEType
finite element types applied during generation of surface element mesh

Available since version 3.

ConversionCoeff : double
triangle-to-quadrilateral conversion coefficient
Available since version 3.

ForcingRatio : IRobotMeshForcingRatio
forcing ratio for the selected type of surface finite elements
Available since version 3.

IRobotMeshDelaunayParams

Parameters of Delaunay's method.
Available since version 3.

Properties:

Type : IRobotMeshDelaunayType
exact type of mesh generation method according to Delaunay's method
Available since version 3.

H_max : double

length of the penultimate wave before the process of increasing mesh elements is completed
Available since version 3.

Q : double
relation of lengths of two successive waves
Available since version 3.

EmittersDefault : bool
Default emitter
Available since version 3.

EmittersUser : bool
User-defined emitter
Available since version 3.

EmittersSmoothing : bool
Smoothing of a finite element mesh
Available since version 3.

H0 : double
first wave length
Available since version 3.

IRobotMeshSurfaceParams

Parameters of surface element meshing
Available since version 3.

Properties:

Method : IRobotMeshMethod
Definition of meshing method
Available since version 3.

Generation : IRobotMeshGeneration
Parameters of mesh generation
Available since version 3.

FiniteElems : IRobotMeshSurfaceFiniteElems
Definition of surface finite elements used during mesh generation
Available since version 3.

Coons : IRobotMeshCoonsParams
parameters of Coons method
Available since version 3.

Delaunay : IRobotMeshDelaunayParams
parameters of Delaunay's method
Available since version 3.

IRobotMeshVolumeParams

Parameters of volumetric element mesh generation.
Available since version 3.

Properties:

MeshDensity : double = 0
refinement degree for volumetric finite element mesh

Available since version 3.

AdditionalSurfaceMeshing : bool
flag switching on additional mesh generation on a solid surface while generating a mesh of volumetric finite elements;
if this option is switched on, it enables additional mesh generation on a solid (surface) contour which will affect the size of mesh elements within a solid
Available since version 3.

FiniteElemsType : IRobotMeshSurfaceFEType
finite element type applied during generation of a volumetric element mesh
Available since version 3.

IRobotMeshParams

Parameters of finite element meshing.
Available since version 3.

Properties:

VolumeParams : IRobotMeshVolumeParams
set of attributes for generation of a volumetric element mesh
Available since version 3.

SurfaceParams : IRobotMeshSurfaceParams
set of attributes for generation of a surface element mesh
Available since version 3.

Flag : IRobotMeshAccessType
flag of access to parameters of surface and volumetric mesh model
Available since version 3.

MeshType : IRobotMeshType
types of generated mesh for surface or volumetric finite elements
Available since version 3.

IRobotMeshAccessType

Availability of interfaces for mesh generation.
Available since version 3.

Properties:

I_MAT_SURFACE_AND_VOLUMETRIC : = 1
flag indicating availability of the interface of surface mesh parameters and the interface of volumetric mesh parameters
Available since version 3.

I_MAT_SURFACE_ONLY : = 2
flag indicating availability only of the interface of surface mesh parameters
Available since version 3.

I_MAT_VOLUMETRIC_ONLY : = 3
flag indicating availability only of the interface of volumetric mesh parameters
Available since version 3.

IRobotMeshRefinementType

Methods of finite element mesh refinement.
Available since version 3.

Properties:

I_MRT_DOUBLE : = 1
each finite element edge is divided into two parts
Available since version 3.

I_MRT_TRIPLE : = 2
each finite element edge is divided into three parts
Available since version 3.

I_MRT_SIMPLE : = 3
Finite element edges will not be divided
Available since version 3.

Complex attributes shared by objects of different types

Material

IRobotMaterialData

Material parameters are described by means of the RobotMaterialdata interface. An object of the type contains data of each label (complex attribute) of the I_LT_BAR_MATERIAL type

Properties:

Name : string
material name

Nuance : string
additional short material description

Type : IRobotMaterialType
material type

Steel_Thermal : bool
thermal flag for steel materials. When it is on, it refers to special processing during production.

Timber_Type : IRobotMaterialTimberType
timber material type (glued multi-layered or normal)

Default : bool
default material (1 - yes, 0 - no)

SecondName : string
second name of the material, available only during editing material database under MS Access

CB71_Category : int
additional characteristics for CD71 code - 3 categories (I, II, II)

CB71_Nature : int
additional characteristics for CD71 code - 2 natures (resinous / non-resinous)

E : double
Young's modulus / axial Young modulus for timber

NU : double
Poisson ratio

LX : double
thermal expansion coefficient

RO : double
material density (mass)

RE : double
yield point (steel, aluminum), compression resistance (concrete)

RE_AxCompr : double
axial compression resistance (timber) (not implemented)

CS : double
reduction coefficient for shear limit stress (steel, aluminium) (not implemented)

RT : double
tension resistance (steel)

E_5 : double
Young's 5% module (timber) only for EC5 code

E_Trans : double
Young transverse module (timber) only for EC5 code

PN_E_Trans : double
Young transverse module (timber) for Polish code (not implemented)

PN_E_Additional : double
additional Young module (timber) for Polish code (not implemented)

RE_Bending : double
bending resistance (timber)

RE_AxTens : double
axial tension resistance (timber)

RE_TrTens : double
transverse tension resistance (timber)

RE_TrCompr : double
transverse compression resistance (timber)

RE_Shear : double
shear resistance (timber)

DumpCoef : double
dumping coefficient

CB71_Retreat : double
additional characteristic for CB71 code - retreat in %

CB71_Humidity : double
additional characteristic for CB71 code - humidity in %

EC_Deformation : double
shear deformation module

PN_Deformation : double
shear deformation module for Polish code

GMean : double
Kirchoff : double
shear (transversal) module

Operations:

LoadFromDBase (material_name : string) : bool
The function allows one to load the data on a material with the defined name to the material database. The returned logical value (True, False) indicates whether the information on the material with the defined name has been found in the material database or not.

SaveToDBase ()
Function saves material to the user database.
Available since version 3.

IRobotMaterialType

Types of materials accepted by Robot.

Properties:

I_MT_STEEL : = 1
steel

I_MT_ALUMINIUM : = 3
aluminum

I_MT_TIMBER : = 4
timber

I_MT_CONCRETE : = 2
concrete

I_MT_OTHER : = 5
other

I_MT_ALL : = 0
identifier denoting all material types
Available since version 1.7.

IRobotMaterialTimberType

Types of timber materials defined by Robot.

Properties:

I_MTT_NORMAL : = 0
normal timber

I_MTT_GLUE_LAMINATED : = 1
multi-layered glued timber

I_MTT_KERTO_S : = 2
Available since version 1.7.

I_MTT_KERTO_Q : = 3
Available since version 1.7.

I_MTT_KERTO_S_OLD : = 4

Available since version 2.5.

I_MTT_KERTO_Q_OLD : = 5

Available since version 2.5.

IRobotMaterialModel

Available material models
Available since version 3.

Properties:

I_MM_ELASTIC : = 0

Available since version 3.

I_MM_HUBER_MISES : = 1

Available since version 3.

I_MM_DRUCKER_PRAGER : = 2

Available since version 3.

I_MM_RANKIN : = 3

Available since version 3.

I_MM_MOHR_COULOMB : = 4

Available since version 3.

IRobotMaterialElasticType

Available models of material elasticity.
Available since version 3.5.

Properties:

I_MET_PERFECTLY_PLASTIC : = 0
elastic-perfectly plastic model
Available since version 3.5.

I_MET_PLASTIC_WITH_HARDENING : = 1
elasto-plastic model with hardening
Available since version 3.5.

IRobotMaterialElasticModel

Model of material elasticity.
Available since version 3.5.

Properties:

Model : IRobotMaterialElasticType

Available since version 3.5.

Coeff : double
coefficient E1 / E for the elasto-plastic model with hardening
Available since version 3.5.

UnloadingMethod : IRobotMaterialElasticUnloadingMethod
unloading method
Available since version 3.5.

UnloadingCoeff : double
factor for the mixed unloading method
Available since version 3.5.

IRobotMaterialElasticUnloadingMethod

Available methods of material unloading.

Available since version 3.5.

Properties:

I_MEUM_ELASTIC : = 1
elastic method of unloading

Available since version 3.5.

I_MEUM_PLASTIC : = 2
plastic method of unloading
Available since version 3.5.

I_MEUM_DAMAGE : = 3
damage method of unloading
Available since version 3.5.

I_MEUM_MIXED : = 4
mixed method of unloading
Available since version 3.5.

Non-linear links

Non-linear connection models.
Available since version 3.

IRobotNonlinearLinkServer

Server of non-linear connection models.
Available since version 3.

Properties:

Count : long
Number of available models
Available since version 3.

Operations:

Get (_idx : long) : IRobotNonlinearLink
Function takes the interface for defining a model of the specified index.
Available since version 3.

Create (_name : string) : IRobotNonlinearLink
Function creates and returns a new non-linear connection model.
Available since version 3.

Find (_name : string) : long
Function returns index of a model of the specified name.

Available since version 3.

Remove (_idx : long)
Function deletes model of the specified index.
Available since version 3.

IRobotNonlinearLink

Definintion of a non-linear connection model.
Available since version 3.

Properties:

Name : string
Model name
Available since version 3.

ModelType : IRobotNonlinearLinkModelType
type of the relationship described by the model
Available since version 3.

Symetry : bool
curve symmetry
Available since version 3.

Operations:

GetParams (_axis_type : IRobotNonlinearLinkSemiAxisType = I_NLSAT_ANY) : IRobotNonlinearLinkParams
Function takes a set of parameters describing a given curve.
Available since version 3.

SetParams (_params : IRobotNonlinearLinkParams, _axis_type : IRobotNonlinearLinkSemiAxisType = I_NLSAT_ANY)
Function updates the set of parameters describing a curve.
Available since version 3.

SetCurveType (_curve_type : IRobotNonlinearLinkCurveType, _axis_type : IRobotNonlinearLinkSemiAxisType = I_NLSAT_ANY)
Function sets the curve type for the indicated semi-axis.
Available since version 3.

GetCurveType (_axis_type : IRoborNonlinearLinkSemiAxisType = I_NLSAT_ANY) : IRobotNonlinearLinkCurveType
Function takes the curve type for the indicated semi-axis.
Available since version 3.

IRobotNonlinearLinkParams

Set of parameters describing any curve.
Available since version 3.

Properties:

CurveType : IRobotNonlinearLinkCurveType
curve type
Available since version 3.

IRobotNonlinearLinkParamsLinear

Derived from IRobotNonlinearLinkParams
Interface specific to the LINEAR curve type.
Available since version 3.

Properties:

K : double

Available since version 3.

IRobotNonlinearLinkParamsBLinear

Derived from IRobotNonlinearLinkParams
Interface specific to the B_LINEAR curve type.
Available since version 3.

Properties:

K1 : double

Available since version 3.

D1 : double

Available since version 3.

K2 : double

Available since version 3.

IRobotNonlinearLinkCurveType

Available curve types:
Available since version 3.

Properties:

I_NLCT_LINEAR : = 1

Available since version 3.

I_NLCT_B_LINEAR : = 2

Available since version 3.

I_NLCT_PARABOLIC : = 3

Available since version 3.

I_NLCT_PARABOLIC_EC2 : = 4

Available since version 3.

I_NLCT_PERFECTLY_PLASTIC : = 5

Available since version 3.

I_NLCT_PLASTIC_WITH_HARDENING : = 6

Available since version 3.

I_NLCT_GAP_HOOK : = 7

Available since version 3.

I_NLCT_CUSTOM : = 8

Available since version 3.

IRobotNonlinearLinkParamsParabolic

Derived from IRobotNonlinearLinkParams
Interface specific to the PARABOLIC and PARABOLIC_EC2 curve types.
Available since version 3.

Properties:

K : double

Available since version 3.

Dlim : double

Available since version 3.

Dmax : double

Available since version 3.

Flim : double

Available since version 3.

IRobotNonlinearLinkParamsPlastic

Derived from IRobotNonlinearLinkParams
Interface specific to the PERFECTLY_PLASTIC curve type.
Available since version 3.

Properties:

K : double

Available since version 3.

Dlim : double

Available since version 3.

Flim : double

Available since version 3.

W : double

Available since version 3.

IRobotNonlinearLinkParamsPlasticHardening

Derived from IRobotNonlinearLinkParams
Interface specific to the PLASTIC_HARDENING curve type.
Available since version 3.

Properties:

K0 : double

Available since version 3.

K1 : double

Available since version 3.

Dlim : double

Available since version 3.

Flim : double

Available since version 3.

W : double

Available since version 3.

IRobotNonlinearLinkParamsGapHook

Derived from IRobotNonlinearLinkParams
Interface specific to the GAP_HOOK curve type.
Available since version 3.

Properties:

K : double

Available since version 3.

D : double

Available since version 3.

IRobotNonlinearLinkParamsCustom

Derived from IRobotNonlinearLinkParams
Interface specific to the CUSTOM curve type; it describes a curve composed of freely defined segments.
Available since version 3.

Properties:

Count : long
number of function segments
Available since version 3.

Operations:

Get (_idx : long) : IRobotNonlinearLinkParamsCustomSegment
Function takes a set of parameters for defining a segment of a given index.
Available since version 3.

New () : IRobotNonlinearLinkParamsCustomSegment
Function creates and returns a new curve segment.
Available since version 3.

Set (_idx : long, _segment : IRobotNonlinearLinkParamsCustomSegment)
Function updates parameters describing a segment of a given index.
Available since version 3.

Remove (_idx : long)
Function deletes a segment of a given index.
Available since version 3.

IRobotNonlinearLinkParamsCustomSegment

Interface defining segment of the CUSTOM type curve
Available since version 3.

Properties:

OriginPoint : double
segment beginning point
Available since version 3.

Expression : string
expression representing the functional relationship
Available since version 3.

Constant : bool
flag indicating if function expression is a constant number (y = a)
Available since version 3.

IRobotNonlinearLinkModelType

Available relation types described by a non-linear model.
Available since version 3.

Properties:

I_NLMT_FORCE_DISPLACEMENT : = 1
force - displacement
Available since version 3.

I_NLMT_MOMENT_ROTATION : = 2
moment - rotation
Available since version 3.

IRobotNonlinearLinkSemiAxisType

Semi-axis types.
Available since version 3.

Properties:

I_NLSAT_POSITIVE : = 1
Positive semi-axis
Available since version 3.

I_NLSAT_NEGATIVE : = 2
Negative semi-axis

I_NLSAT_ANY : = 3
Arbitrary semi-axis
Available since version 3.

IRobotNonlinearLinkMngr

Support for relations of the non-linear model - direction type.
Available since version 3.

Operations:

Set (_direction : IRobotDegreeOfFreedom, _model_name : string)
Function determines a non-linear model for the specified drection.
Available since version 3.

Get (_direction : IRobotDegreeOfFreedom) : string
Function returns the name of a non-linear model assigned to the specified direction.
Available since version 3.

IsDefined (_direction : IRobotDegreeOfFreedom) : bool
Function checks a non-zero value (True), if a non-linear model has been assigned to the indicated direction.
Available since version 3.

Delete (_direction : IRobotDegreeOfFreedom)
Function deletes a non-linear model from the indicated direction.
Available since version 3.

Results

IRobotResultServer

The server makes available all types of results obtained by performing structure calculations.

Properties:

Available : bool

flag indicating if the results are available and up to date

Nodes : IRobotNodeResultServer
server of all results for nodes

Bars : IRobotBarResultServer
server of all results for bars

Advanced : IRobotAdvancedResultServer
Available since version 1.7.

FiniteElems : IRobotFeResultServer
result server for finite elements
Available since version 2.5.

Extremes : IRobotExtremeResultServer
server providing access to some extreme values
Available since version 3.5.

Any : IRobotUniversalResultAccess
access to any calculation result
Available since version 3.5.

Total : IRobotStructureValues
physical properties of a whole structure
Available since version 7.5.

IRobotUniversalResultAccess

Universal access interface for calculation results.
In order to obtain the relevant result the object should be first appropriately parametrized by determining what result type, for which load case and for which structure element is to be obtained, and then ResultValue should be taken.

Properties:

ResultId : long
identifier determining the result type which is to be taken

Node : long
node number for which the result is to be taken

Bar : long
bar number for which the result is to be taken

DivCount : long
number of bar division points (only for the results obtained in the defined point of bar division)

DivPoint : long
number of bar division point (only for the results obtained in the defined point of bar division)

RelativePoint : double
relative point on the bar (significant only for the results obtained in the defined bar point)

LoadCase : long
number of the load case for which the result is to be taken

LoadCaseCmpnt : long
number of the load case component (significant for cases including more than one component, e.g. code combination)

Mode : long
number of the load case mode (significant for cases for which many modes are generated)

ModeCmb : IRobotModeCombinationType
type of mode combination

Available : bool
flag indicating if the result for defined parameters can be obtained

ResultValue : double
result value for defined parameters (provided that the Available flag assumes value different from zero - True)

Panel : long
panel number
Available since version 3.5.

Element : long
number of the finite element
Available since version 3.5.

Layer : IRobotFeLayerType
layer
Available since version 3.5.

LayerArbitraryValue : double
layer thickness
Available since version 3.5.

ReinforceCalcMethod : IRobotReinforceCalcMethods
calculation method for reinforcement
Available since version 3.5.

ReducedCutPos : IRobotFeResultReducedCutPosition
position of the panel cut (for reduced results)
Available since version 3.5.

ResultType : IRobotUniversalResultType
result type

Storey : string
story name

CalcPoint : long

Available since version 8.

Operations:

Reset ()
Function sets as zero all the settings concerning the result type and the manner of its taking.

SetDirX (_def_type : IRobotObjLocalXDirDefinitionType, _x : double, _y : double, _z : double)
Function allows defining direction of X axis of the system in which results will be presented. The function should be activated only when the vector of detailed results is being taken. Function parameters determine coordinates of the direction vector of X axis in the Cartesian system or coordinates of the central point in the Polar system. If all the values: x, y, z are set as zeros, a direction compatible with the local system of the selected panel will be defined.
Available since version 3.5.

GetDirX (_x : double*, _y : double*, _z : double*) : IRobotObjLocalXDirDefinitionType
Function takes direction of X axis of the local coordinate system in which results will be presented. The values read are the coordinates of the direction vector of X axis for definition in the Cartesian system or the coordinates of the central point for definition in the Polar system. If values of all the arguments: x, y, z are set as zeros, then the system direction is compatible with the local panel system. In addition, the function returns a type of system direction definition.
Available since version 3.5.

ResultValue3D (_val1 : double*, _val2 : double*, _val3 : double*)
Function returns a result in a form of a triple of numbers.

LinearSupports : bool
Linear supports
Available since version 8.1.

IRobotExtremeResultServer

Server of extreme results.
Available since version 3.5.

Operations:

MinValue (_params : IRobotExtremeParams) : IRobotExtremeValue
Function returns the minimal value calculated for the parameters given.
Available since version 3.5.

MaxValue (_params : IRobotExtremeParams) : IRobotExtremeValue
Function returns the maximal value calculated for the parameters given.
Available since version 3.5.

IRobotUniversalResultType

Types of results made available by the RobotUniversalResultAccess object.

Properties:

I_URT_VALUE : = 1
result has a form of a single value

I_URT_VALUE_3D : = 2
result has a form of a triple of numbers

I_URT_UNKNOWN : = 0

IRobotStructureValues

Physical properties for the structure model.
Available since version 7.5.

Operations:

GetMass (_case_num : long) : double
Function returns a mass of the whole structure for the given load case.
Available since version 7.5.

GetG (_case_num : long) : IRobotGeoPoint3D

Function returns coordinates of structure center of gravity for the given load case.
Available since version 7.5.

GetT (_case_num : long) : IRobotGeoPoint3D
Function returns coordinates of structure center of torsion for the given load case.
Available since version 7.5.

GetIx (_case_num : long) : double
Function returns mass moment of inertia according to X axis for the given load case.
Available since version 7.5.

GetIy (_case_num : long) : double
Function returns mass moment of inertia according to Y axis for the given load case.
Available since version 7.5.

GetIz (_case_num : long) : double
Function returns mass moment of inertia according to Z axis for the given load case.
Available since version 7.5.

GetEx2 (_case_num : long) : double
Function returns additional eccentricity according to X axis for the given load case.
Available since version 7.5.

GetEy2 (_case_num : long) : double
Function returns additional eccentricity according to Y axis for the given load case.
Available since version 7.5.

Structures containing complex calculation results

IRobotReactionData

The structure describing the value of reactions.

Properties:

FX : double
FY : double
FZ : double
MX : double
MY : double
MZ : double

IRobotDisplacementData

Structure describing the value of displacements.

Properties:

UX : double
UY : double
UZ : double
RX : double
RY : double
RZ : double

IRobotBarForceData

Structure describing values of forces, moments and elastic ground reaction for a bar.

Properties:

FX : double
FY : double
FZ : double
MX : double
MY : double
MZ : double
KYAvailable : bool
flag indicating availability of the value of elastic ground reaction KY
Available since version 3.5.

KZAvailable : bool
flag indicating availability of the value of elastic ground reaction KZ
Available since version 3.5.

KY : double

Available since version 3.5.

KZ : double

Available since version 3.5.

IRobotBarStressData

A structure includes an information about stresses in a selected bar point for a selected load case.

Properties:

Smax : double
Smin : double
SmaxMY : double
SminMY : double
SmaxMZ : double

SminMZ : double
FXSX : double
ShearY : double
ShearZ : double
Torsion : double

IRobotBarDeflectionData

A type of data describing deflections for the selected bar and case in a given bar point.

Properties:

UX : double
UY : double
UZ : double
PosUX : double
a relative point position on a bar for which UX value was read

PosUY : double
a relative point position for which UY value was read

PosUZ : double
a relative point position for which UZ value was read

IRobotBarBucklingData

The structure describing the eigenvalues for a single member, load case and mode.

Properties:

CriticalCoef : double
critical coefficient

CriticalForce : double
critical force

BuckLengthY : double
buckling length Y

BuckLengthZ : double
buckling length Z

SlendY : double
slenderness Y

SlendZ : double
slenderness Z

IRobotEigenvalues

Data structure containing eigenvalues.

Properties:

EigenValue : double
Available since version 1.7.

Frequence : double
Available since version 1.7.

Period : double
Available since version 1.7.

Pulsation : double
Available since version 1.7.

Damping : double
Available since version 1.7.

Precision : double
Available since version 1.7.

Energy : double
Available since version 1.7.

AvPartCoeff : double
Available since version 1.7.

IRobotForcesData

Data structure containing force and moment values.

Properties:

FX : double
Available since version 1.7.

FY : double
Available since version 1.7.

FZ : double
Available since version 1.7.

MX : double
Available since version 1.7.

MY : double
Available since version 1.7.

MZ : double
Available since version 1.7.

IRobotTimeHistoryResults

Results for time history analysis
Available since version 3.

Properties:

VX : double
Value of velocity component VX
Available since version 3.

VY : double
Value of velocity component VY
Available since version 3.

VZ : double
Value of velocity component VZ
Available since version 3.

VRX : double
Value of velocity component VRX
Available since version 3.

VRY : double
Value of velocity component VRY
Available since version 3.

VRZ : double
Value of velocity component VRZ
Available since version 3.

AX : double
Value of acceleration component AX
Available since version 3.

AY : double
Value of acceleration component AY
Available since version 3.

AZ : double
Value of acceleration component AZ
Available since version 3.

ARX : double
Value of acceleration component ARX
Available since version 3.

ARY : double
Value of acceleration component ARY
Available since version 3.

ARZ : double
Value of acceleration component ARZ
Available since version 3.

Time : double
time corresponding to the specified step
Available since version 3.

IRobotExtremeValue

Strucuture providing access to information about the extreme value.
Available since version 3.5.

Properties:

Value : double
value
Available since version 3.5.

Case : long
number of the load case for which the specified extreme value has been obtained
Available since version 3.5.

CaseCmpnt : long
additional number indicating a load case component
- for a code combination, it is a number of its component
- for modal analysis, it is a mode number
- for results of time history analysis, it is a step number
Available since version 3.5.

Node : long
number of the node in which the extreme value has been obtained
Available since version 3.5.

Position : double
relative position on a bar in which the extreme value has been obtained
Available since version 3.5.

Bar : long
number of the bar for which the extreme value has been obtained

Available since version 3.5.

ModeCmb : IRobotModeCombinationType

designation of the mode combination for which the specified extreme value has been obtained (significant only when CaseCmpnt component equals zero)
Available since version 3.5.

IsAvailable : bool
flag indicating if the extreme value has been taken correctly from the result server
Available since version 3.5.

IRobotExtremeParams

Strucuture containing parameters for which the extreme value is searched.
Available since version 3.5.

Properties:

Selection : IRobotMultiSelection
selection of structure components (nodes, bars, load cases) that should be considered while searching the extreme value
Available since version 3.5.

ValueType : IRobotExtremeValueType
type of the result for which the extreme value is to be taken
Available since version 3.5.

BarDivision : long
number of bar divisions - significant only for bar results
Available since version 3.5.

IRobotExtremeValueType

Set of results for which the extreme values are made accessible.
Available since version 3.5.

Properties:

I_EVT_REACTION_FX : = 109

Available since version 3.5.

I_EVT_REACTION_FY : = 110

Available since version 3.5.

I_EVT_REACTION_FZ : = 111

Available since version 3.5.

I_EVT_REACTION_MX : = 112

Available since version 3.5.

I_EVT_REACTION_MY : = 113

Available since version 3.5.

I_EVT_REACTION_MZ : = 114

Available since version 3.5.

I_EVT_DISPLACEMENT_NODE_UX : = 141

Available since version 3.5.

I_EVT_DISPLACEMENT_NODE_UY : = 142

Available since version 3.5.

I_EVT_DISPLACEMENT_NODE_UZ : = 143

Available since version 3.5.

I_EVT_DISPLACEMENT_NODE_RX : = 144

Available since version 3.5.

I_EVT_DISPLACEMENT_NODE_RY : = 145

Available since version 3.5.

I_EVT_DISPLACEMENT_NODE_RZ : = 146

Available since version 3.5.

I_EVT_DEFLECTION_UX : = 153

Available since version 3.5.

I_EVT_DEFLECTION_UY : = 154

Available since version 3.5.

I_EVT_DEFLECTION_UZ : = 155

Available since version 3.5.

I_EVT_DISPLACEMENT_BAR_UX : = 147

Available since version 3.5.

I_EVT_DISPLACEMENT_BAR_UY : = 148

Available since version 3.5.

I_EVT_DISPLACEMENT_BAR_UZ : = 149

Available since version 3.5.

I_EVT_DISPLACEMENT_BAR_RX : = 150

Available since version 3.5.

I_EVT_DISPLACEMENT_BAR_RY : = 151

Available since version 3.5.

I_EVT_DISPLACEMENT_BAR_RZ : = 152

Available since version 3.5.

I_EVT_FORCE_BAR_FX : = 165

Available since version 3.5.

I_EVT_FORCE_BAR_FY : = 166

Available since version 3.5.

I_EVT_FORCE_BAR_FZ : = 167

Available since version 3.5.

I_EVT_FORCE_BAR_MX : = 168

Available since version 3.5.

I_EVT_FORCE_BAR_MY : = 169

Available since version 3.5.

I_EVT_FORCE_BAR_MZ : = 170

Available since version 3.5.

I_EVT_STRESS_BAR_SMAX : = 171

Available since version 3.5.

I_EVT_STRESS_BAR_SMIN : = 172

Available since version 3.5.

I_EVT_STRESS_BAR_SMAX_MY : = 174

Available since version 3.5.

I_EVT_STRESS_BAR_SMAX_MZ : = 175

Available since version 3.5.

I_EVT_STRESS_BAR_SMIN_MY : = 176

Available since version 3.5.

I_EVT_STRESS_BAR_SMIN_MZ : = 177

Available since version 3.5.

I_EVT_STRESS_BAR_FX_SX : = 173

Available since version 3.5.

I_EVT_STRESS_BAR_TY : = 179

Available since version 3.5.

I_EVT_STRESS_BAR_TZ : = 180

Available since version 3.5.

I_EVT_STRESS_BAR_T : = 182

Available since version 3.5.

I_EVT_PSEUDOSTATIC_FORCE_FX : = 251

Available since version 3.5.

I_EVT_PSEUDOSTATIC_FORCE_FY : = 252

Available since version 3.5.

I_EVT_PSEUDOSTATIC_FORCE_FZ : = 253

Available since version 3.5.

I_EVT_PSEUDOSTATIC_FORCE_MX : = 254

Available since version 3.5.

I_EVT_PSEUDOSTATIC_FORCE_MY : = 255

Available since version 3.5.

I_EVT_PSEUDOSTATIC_FORCE_MZ : = 256

Available since version 3.5.

I_EVT_PSEUDOSTATIC_FORCE_GX : = 1328

Available since version 3.5.

I_EVT_PSEUDOSTATIC_FORCE_GY : = 1329

Available since version 3.5.

I_EVT_PSEUDOSTATIC_FORCE_GZ : = 1330

Available since version 3.5.

I_EVT_PSEUDOSTATIC_FORCE_TX : = 1331

Available since version 3.5.

I_EVT_PSEUDOSTATIC_FORCE_TY : = 1332

Available since version 3.5.

I_EVT_PSEUDOSTATIC_FORCE_TZ : = 1333

Available since version 3.5.

I_EVT_TIME_VX : = 257

Available since version 3.5.

I_EVT_TIME_VY : = 258

Available since version 3.5.

I_EVT_TIME_VZ : = 259

Available since version 3.5.

I_EVT_TIME_VRX : = 260

Available since version 3.5.

I_EVT_TIME_VRY : = 261

Available since version 3.5.

I_EVT_TIME_VRZ : = 262

Available since version 3.5.

I_EVT_TIME_AX : = 263

Available since version 3.5.

I_EVT_TIME_AY : = 264

Available since version 3.5.

I_EVT_TIME_AZ : = 265

Available since version 3.5.

I_EVT_TIME_ARX : = 266

Available since version 3.5.

I_EVT_TIME_ARY : = 267

Available since version 3.5.

I_EVT_TIME_ARZ : = 268

Available since version 3.5.

Calculation results for nodes

IRobotNodeResultServer

All types or results for nodes are made accessible by the appropriate server of the RobotNodeResultServer type.

Properties:

Reactions : IRobotReactionServer
server provides access to reaction values in nodes.

Displacements : IRobotNodeDisplacementServer
server provides access to nodal displacements

Buckling : IRobotNodeBucklingServer
the server of results of buckling analysis for nodes

PseudostaticForces : IRobotPseudostaticForceServer

server providing access to values of pseudostatic forces
Available since version 1.7.

IRobotReactionServer

The server providing access to values of reactions for nodes.

Operations:

Value (node : long, case_num : long) : IRobotReactionData
The function returns a structure containing the values of reactions for a given node and load case.

Local (node : long, case_num : long) : IRobotReactionData
The function returns a structure containing reaction values in the local coordinate system for the indicated node and load case.

DDC (node : long, case_num : long) : IRobotReactionData
The function returns a structure containing Load Report reaction values for the indicated node and load case

DDCLocal (node : long, case_num : long) : IRobotReactionData
The function returns a structure containing Load Report reaction values for the indicated node and load case

Sum (case_num : long) : IRobotReactionData
The function returns a structure containing values of reaction sums for the indicated load case

DDCSum (case_num : long) : IRobotReactionData
The function returns a structure containing values of reaction sums of the Load Report type for the indicated load case.

SumForce (case_num : long) : IRobotReactionData
The function returns a structure containing the value of sums of forces for the indicated load case.

ValueEx (node_num : long, case_num : long, case_cmpnt : long = 1) :
IRobotReactionData
Function returns the structure containing reaction values for the specified node and load case.
Available since version 1.7.

LocalEx (node : long, _case : long, case_cmpnt : long = 1) : IRobotReactionData
Available since version 1.7.

DDCEx (node : long, _case : long, case_cmpnt : long = 1) : IRobotReactionData
Available since version 1.7.

DDCLocalEx (node : long, _case : long, case_cmpnt : long = 1) : IRobotReactionData
Available since version 1.7.

SumEx (_case : long, case_cmpnt : long = 1) : IRobotReactionData
Available since version 1.7.

DDCSumEx (_case : long, case_cmpnt : long = 1) : IRobotReactionData
Available since version 1.7.

SumForceEx (_case : long, case_cmpnt : long = 1) : IRobotReactionData
Available since version 1.7.

DynValue (node : long, _case : long, mode : long) : IRobotReactionData
Available since version 1.7.

DynLocal (node : long, _case : long, mode : long) : IRobotReactionData
Available since version 1.7.

DynDDC (node : long, _case : long, mode : long) : IRobotReactionData
Available since version 1.7.

DynDDCLocal (node : long, _case : long, mode : long) : IRobotReactionData
Available since version 1.7.

DynSum (_case : long, mode : long) : IRobotReactionData
Available since version 1.7.

DynDDCSum (_case : long, mode : long) : IRobotReactionData
Available since version 1.7.

DynSumForce (_case : long, mode : long) : IRobotReactionData
Available since version 1.7.

DynCombValue (node : long, _case : long, mode_cmb : IRobotModeCombinationType) : IRobotReactionData
Available since version 1.7.

DynCombLocal (node : long, _case : long, mode_cmb : IRobotModeCombinationType) : IRobotReactionData
Available since version 1.7.

DynCombDDC (node : long, _case : long, mode_cmb : IRobotModeCombinationType) : IRobotReactionData
Available since version 1.7.

DynCombDDCLocal (node : long, _case : long, mode_cmb : IRobotModeCombinationType) : IRobotReactionData
Available since version 1.7.

DynCombSum (_case : long, mode_cmb : IRobotModeCombinationType) : IRobotReactionData
Available since version 1.7.

DynCombDDCSum (_case : long, mode_cmb : IRobotModeCombinationType) : IRobotReactionData
Available since version 1.7.

DynCombSumForce (_case : long, mode_cmb : IRobotModeCombinationType) : IRobotReactionData
Available since version 1.7.

IRobotNodeDisplacementServer

The server provides access to all values of nodal displacements.

Operations:

Value (node_num : long, case_num : long) : IRobotDisplacementData
The function returns an object (set of data) describing displacements in the indicated node for the indicated load case.

ValueEx (node : long, _case : long, case_cmpnt : long) : IRobotDisplacementData
Available since version 1.7.

DynValue (node_num : long, case_num : long, _mode : long) : IRobotDisplacementData
Function returns the data structure describing displacements in the defined node for a specified load case and specified mode.
Available since version 1.7.

DynCombValue (node_num : long, case_num : long, mode_cmb : IRobotModeCombinationType) : IRobotDisplacementData
Function returns the data structure containing displacements for the specified node and defined mode combination of a selected load case.
Available since version 1.7.

IRobotNodeBucklingServer

The server provides results of buckling analysis for nodes.

Operations:

EigenVector (node : long, case : long, mode : long) : IRobotDisplacementData
The function returns the displacement vector for the indicated node, load case and mode.

EigenVectorCmb (node : long, _case : long, mode_cmb : IRobotModeCombinationType) : IRobotDisplacementData
Available since version 1.7.

IRobotPseudostaticForceServer

Server providing access to pseudostatic forces.

Operations:

Value (node_num : long, case_num : long, mode_num : long) : IRobotForcesData
Function returns pseudostatic force values in a specified node for a defined load case and mode.
Available since version 1.7.

CombValue (node_num : long, case_num : long, mode_cmb : IRobotModeCombinationType) : IRobotForcesData
Function returns pseudostatic force values in a specified node for a defined load case and mode.
Available since version 1.7.

Calculation results for bars

IRobotBarResultServer

Server makes accessible all types of calculation results for bars.

Properties:

Forces : IRobotBarForceServer
internal forces

Deflections : IRobotBarDeflectionServer
deflections

Stresses : IRobotBarStressServer
stresses

Buckling : IRobotBarBucklingServer
server of results of buckling analysis for bars

Displacements : IRobotBarDisplacementServer
displacement server

Available since version 2.5.

Operations:

GeoImperfections (_bar : long) : IRobotBarDeflectionData
Function returns calculated deflection values in individual directions for the indicated bar.
Available since version 3.

IRobotBarForceServer

Server provides access to values of forces and moments for bars.

Operations:

Value (bar_num : long, case_num : long, point : double) : IRobotBarForceData
The function returns a structure containing the values of all forces and moments for the indicated bar, load case and point along a bar.

ValueByNPoints (bar_num : long, case_num : long, no_points : int, point : int) : IRobotBarForceData
The function returns a structure containing the values of all forces and moments for the indicated bar, load case at the indicated division point.

ValueEx (bar_num : long, case_num : long, case_cmpnt : long, point : double) : IRobotBarForceData
Available since version 1.7.

ValueByNPointsEx (bar_num : long, case_num : long, case_cmpnt : long, no_points : int, point : int) : IRobotBarForceData
Available since version 1.7.

DynValue (bar_num : long, case_num : long, _mode : long, _point : double) : IRobotBarForceData
Function returns the structure containing values of all forces and moments for a specified bar, mode of the load case and point on the bar.
Available since version 1.7.

DynValueByNPoints (bar_num : long, case_num : long, _mode : long, no_points : int, _point : int) : IRobotBarForceData
Function returns the structure containing values of all forces and moments for a specified bar, mode of the load case and point on the bar.
Available since version 1.7.

DynCombValue (bar_num : long, case_num : long, mode_cmb : IRobotModeCombinationType, _point : double) : IRobotBarForceData
Available since version 1.7.

DynCombValueByNPoints (bar_num : long, case_num : long, mode_cmb : IRobotModeCombinationType, no_points : int, _point : int) : IRobotBarForceData
Available since version 1.7.

IRobotBarDeflectionServer

The server makes deflection values for bars available.

Operations:

Value (bar_num : long, case_num : long, pos : double) : IRobotBarDeflectionData
The function gives back a data stucture containing deflection values for a given bar, load case and a point on a bar.

MaxValue (bar_num : long, case_num : long) : IRobotBarDeflectionData
The function gives back maximum deflection values for a given bar, load case and appropriate positions on a bar.

DynValue (bar_num : long, case_num : long, mode_num : long, pos : double) : IRobotBarDeflectionData
The function gives back deflection values in dynamics.

DynCombValue (bar_num : long, case_num : long, pos : double, cmb : IRobotModeCombinationType) : IRobotBarDeflectionData
The function makes available the deflection values for an indicated combination.

DynMaxValue (bar_num : long, case_num : long, mode_num : long) : IRobotBarDeflectionData
The function makes maximum deflection values and appropriate positions on a bar in dynamics available.

DynCombMaxValue (bar_num : long, case_num : long, cmb : IRobotModeCombinationType) : IRobotBarDeflectionData
The function makes available the maximum deflections for the square combination and appropriate positions on a bar.

ValueEx (bar_num : long, case_num : long, case_cmpnt : long, pos : double) : IRobotBarDeflectionData
Available since version 1.7.

MaxValueEx (bar_num : long, case_num : long, case_cmpnt : long) : IRobotBarDeflectionData
Available since version 1.7.

IRobotBarStressServer

Server allows to get stress values for bars.

Operations:

Value (bar_num : long, case_num : long, pos : double) : IRobotBarStressData
The function makes the stress values for a given bar, load case and relative position on a bar available.

ValueEx (bar_num : long, case_num : long, case_cmpnt : long, pos : double) : IRobotBarStressData
Available since version 1.7.

DynValue (bar_num : long, case_num : long, mode_num : long, pos : double) : IRobotBarStressData
Function returns stress values for a selected load case and mode, in the selected point on the bar.
Available since version 1.7.

DynCombValue (bar_num : long, case_num : long, mode_cmb : IRobotModeCombinationType, pos : double) : IRobotBarStressData
Function returns stress values for a selected load case and specified mode combination, in the selected point on the bar.
Available since version 1.7.

IRobotBarBucklingServer

The server provides results of buckling analysis for bars.

Operations:

CriticalCoef (case_num : long, mode_num : long) : double
The function returns the value of the critical coefficient for the indicated load case and mode.

EigenValue (bar : long, case : long, mode : long) : IRobotBarBucklingData
The function returns eigenvalues for the indicated bar, load case and mode.

IRobotBarDisplacementServer

Server providing access to displacement results for bars.
Available since version 2.5.

Operations:

Value (_bar_num : long, _pos : double, _case_num : long, _case_cmpnt : long = 1) : IRobotDisplacementData
Function returns the structure containing displacement values for the indicated bar and load case.
Available since version 2.5.

DynValue (_bar_num : long, _pos : double, _case_num : long, _mode : long) : IRobotDisplacementData
Function returns the structure containing displacement values for the indicated bar, load case and mode.
Available since version 2.5.

DynCombValue (_bar_num : long, _pos : double, _case_num : long, _mode_cmb : IRobotModeCombinationType) : IRobotDisplacementData
Function returns the structure containing displacement values for the indicated bar, load case and mode combinations.
Available since version 2.5.

Dynamic results

IRobotAdvancedResultServer

Properties:

Eigenvalues : IRobotEigenvaluesServer
Available since version 1.7.

MassSum : IRobotMassSumServer
server providing access to sum of masses
Available since version 1.7.

SpectralCoeffs : IRobotSpectralCoefficients
server providing access to spectral coefficients
Available since version 1.7.

Eigenvectors : IRobotEigenvectorsServer
server providing access to eigenvector values
Available since version 1.7.

TimeHistory : IRobotTimeHistoryResultServer
server providing access to results of time history analysis
Available since version 3.

IRobotEigenvaluesServer

Server providing access to eigenvalues.

Operations:

Value (case_num : long, mode_num : long) : IRobotEigenvalues
Available since version 1.7.

CombValue (case_num : long, mode_cmb : IRobotModeCombinationType) : IRobotEigenvalues
Available since version 1.7.

IRobotMassSumServer

Operations:

Current (case_num : long, mode_num : long) : IRobotDisplacementData
Function returns values of participation masses expressed in percentages in the current mode for a specified case and mode, for individual directions.
Available since version 1.7.

Relative (case_num : long, mode_num : long) : IRobotDisplacementData
Function returns values of participation mass sum expressed in percentages from mode 1 to the current mode for a specified case and mode, for individual directions.
Available since version 1.7.

Total (case_num : long, mode_num : long) : IRobotDisplacementData
Function returns values of total participation mass sum in a specified case for a specified mode, for individual directions.
Available since version 1.7.

PartCoeff (case_num : long, mode_num : long) : IRobotDisplacementData
Function returns values of modal participation coefficients in a specified case for a defined mode, for individual directions.
Available since version 1.7.

IRobotSpectralCoefficients

Server provides access to spectral coefficient values.

Operations:

SpectrCoef (case_num : long, mode_num : long) : double
Function returns spectral coefficient value for a specified case and mode.
Available since version 1.7.

PartCoef (case_num : long, mode_num : long) : IRobotDisplacementData
Function returns values of spectral participation coefficients in a specified case for a defined mode, for individual directions.
Available since version 1.7.

ModeCoef (case_num : long, mode_num : long) : IRobotDisplacementData
Function returns values of spectral mode coefficients in a specified case and mode for individual directions.
Available since version 1.7.

IRobotEigenvectorsServer

Server providing access to eigenvectors.

Operations:

Value (node_num : long, case_num : long, mode_num : long) : IRobotDisplacementData
Function returns eigenvector for a specified node, load case and mode.
Available since version 1.7.

CombValue (node_num : long, case_num : long, mode_cmb : IRobotModeCombinationType) : IRobotDisplacementData
Function returns eigenvector for a specified node, load case and mode combination.
Available since version 1.7.

IRobotTimeHistoryResultServer

Result server for time history analysis.
Available since version 3.

Operations:

Value (_node : long, _case : long, _step : long) : IRobotTimeHistoryResults
Function returns a set of time history analysis results for a given node, case and step.
Available since version 3.

Calculation results for finite elements

Available since version 2.5.

IRobotFeLayerType

Available designations of layers for which calculation results for planar finite elements may be obtained.
Available since version 2.5.

Properties:

I_FLT_UPPER : = 1
top layer
Available since version 2.5.

I_FLT_MIDDLE : = 2
middle layer
Available since version 2.5.

I_FLT_LOWER : = 3

bottom layer
Available since version 2.5.

I_FLT_ARBITRARY : = 4
relative layer width defined arbitrarily
Available since version 2.5.

IRobotFeResultParams

Structure defining parameters necessary to take calculation results for finite elements. The type of returned values is strictly determined by the setting configuration of the fields: Panel, Node, Element. The following variants are possible:
1. the Node node number is set
globally averaged results at a node are returned
2. the Node node number and the Panel panel number are set
results at a node averaged within a given panel are the results returned
3. the Node node number and the Element finite element number are set
exact results at a node are returned
4. the Element finite element number is set
results at a center of the indicated element are the results returned

Available since version 2.5.

Properties:

Panel : long
panel number
Available since version 2.5.

Node : long
node number
Available since version 2.5.

Case : long
load case number
Available since version 2.5.

Layer : IRobotFeLayerType
layer
Available since version 2.5.

CaseCmpnt : long
component number for a complex component case or mode number for a dynamic case
Available since version 2.5.

Element : long
finite element number

Available since version 2.5.

LayerArbitraryValue : double
layer thickness
Available since version 2.5.

ModeCmb : IRobotModeCombinationType
type of mode combination for a dynamic case
Available since version 2.5.

CalcMethod : IRobotReinforceCalcMethods
type of reinforcement calculation method
Available since version 2.5.

Operations:

SetDirX (_def_type : IRobotObjLocalXDirDefinitionType, _x : double, _y : double, _z : double)
The function enables defining the X axis direction of the system in which the results will be presented. Activation of the function should take place only when the vector of detailed results is being taken. Function parameters determine the coordinates of the direction vector of the X axis in the Cartesian system or the coordinates of the central point in the polar system. In case when all the values: x, y, z are set as zeros, a direction compatible with the local system of the selected panel will be defined.
Available since version 2.5.

GetDirX (_x : double*, _y : double*, _z : double*) : IRobotObjLocalXDirDefinitionType
Function takes the direction of the local X axis of the coordinate system in which the results will be presented. The values read are the coordinates of the direction vector of the X axis for definition in the Cartesian system or the coordinates of the central point for definition in the polar system. In case when values of all the arguments: x, y, z are set as zeros, then the system direction is compatible with the panel local system. In addition, the function returns a type of the system direction definition.
Available since version 2.5.

IRobotFeResultDetailed

Detailed result vector.
Available since version 2.5.

Properties:

SXX : double
sXX stress value
Available since version 2.5.

SYY : double
sYY stress value
Available since version 2.5.

SXY : double
sXY stress value
Available since version 2.5.

NXX : double
NXX membrane force value
Available since version 2.5.

NYY : double
NYY membrane force value

Available since version 2.5.

NXY : double
NXY membrane force value

Available since version 2.5.

MXX : double
MXX moment value
Available since version 2.5.

MYY : double
MYY moment value
Available since version 2.5.

MXY : double
MXY moment value

Available since version 2.5.

TXX : double
tXX shear stress value
Available since version 2.5.

TYY : double
tYY shear stress value
Available since version 2.5.

QXX : double
QXX shear force value
Available since version 2.5.

QYY : double
QYY shear force value
Available since version 2.5.

UXX : double
UXX displacement
Available since version 2.5.

UYY : double
UYY displacement

Available since version 2.5.

WNorm : double
WNorm displacement

Available since version 2.5.

RXX : double
RXX rotation
Available since version 2.5.

RYY : double
RYY rotation
Available since version 2.5.

RNorm : double
RNorm rotation
Available since version 2.5.

PNorm : double
pNorm soil reaction value
Available since version 2.5.

IRobotFeResultServer

Result server for planar finite elements.
Available since version 2.5.

Properties:

PanelCuts : IRobotPanelCutMngr
manager of panel cuts
Available since version 3.

Operations:

Detailed (_param : IRobotFeResultParams) : IRobotFeResultDetailed
Function returns the detailed result vector corresponding to the parameters defined in the argument.
Available since version 2.5.

Principal (_param : IRobotFeResultParams) : IRobotFeResultPrincipal
Function returns the principal result vector corresponding to the parameters defined in the argument.
Available since version 2.5.

Complex (_param : IRobotFeResultParams) : IRobotFeResultComplex

Function returns a set of complex results corresponding to the parameters defined in the argument.
Available since version 3.

Reinforcement (_elem_or_panel : long, _node : long = 0) : IRobotFeResultReinforcement
Function returns a result vector for the reinforcement corresponding to the specified parameters. There are two variants of obtaining results by the appropriate setting of arguments:
1. first argument determines number of a panel, second argument determines node number
2. first argument determines number of a finite element, while second argument is set as zero
Available since version 3.

Reduced (_panel : long, _cut : IRobotFeResultReducedCutPosition, _case : long) : IRobotFeResultReduced
Function returns the vector of reduced results for panels corresponding to the specified parameters.
Available since version 3.

MinValue (_params : IRobotFeExtremeParams) : IRobotFeExtremeValue

Available since version 4.

MaxValue (_params : IRobotFeExtremeParams) : IRobotFeExtremeValue

Available since version 4.

MultiMinValue (_params : IRobotFeExtremeParams, _result_ids : IRobotFeMultiResultType) : IRobotFeMultiExtremeValue
Function searches and returns the minimal values for all the specified types of results. Parameters of search for the minimal values are common for all the indicated types of results.
Available since version 4.1.

MultiMaxValue (_params : IRobotFeExtremeParams, _result_ids : IRobotFeMultiResultType) : IRobotFeMultiExtremeValue
Function searches and returns the maximal values for all the specified types of results. Parameters of search for the maximal values are common for all the indicated types of results.
Available since version 4.1.

ReducedEx (_panel : long, _part_idx : long, _cut_pos : IRobotFeResultReducedCutPosition, _case : long, _case_cmpnt : long = 1) : IRobotFeResultReduced
Function returns a vector of reduced results for panels, corresponding to the specified parameters.
Available since version 5.5.

IRobotReinforceCalcMethods

Types of the reinforcement calculation hypotheses.
Available since version 2.5.

Properties:

I_RCM_NEN : = 1
NEN hypothesis
Available since version 2.5.

I_RCM_WOOD_ARMER : = 2
WOOD&ARMER hypothesis
Available since version 2.5.

I_RCM_ANALYTICAL : = 3

Available since version 5.5.

IRobotFeResultPrincipal

Principal result vector.
Available since version 2.5.

Properties:

S1 : double
value of s1 stress
Available since version 2.5.

S2 : double
value of s2 stress

Available since version 2.5.

S1_2 : double
value of s(1-2) stress
Available since version 2.5.

SAL : double
Sal angle
Available since version 2.5.

N1 : double
value of N1 membrane force
Available since version 2.5.

N2 : double
value of N2 membrane force
Available since version 2.5.

N1_2 : double
value of N(1-2) membrane force
Available since version 2.5.

NAL : double
Nal angle
Available since version 2.5.

M1 : double
value of M1 moment
Available since version 2.5.

M2 : double
value of M2 moment
Available since version 2.5.

M1_2 : double
value of M(1-2) moment
Available since version 2.5.

MAL : double
Mal angle
Available since version 2.5.

T1_2 : double
value of T(1-2) shear stress
Available since version 2.5.

Q1_2 : double
value of Q(1-2) shear force
Available since version 2.5.

UGX : double
value of UGX global displacement
Available since version 2.5.

UGY : double
value of UGY global displacement
Available since version 2.5.

UGZ : double
value of UGZ global displacement
Available since version 2.5.

U : double
value of U total displacement
Available since version 2.5.

IRobotFeResultComplex

Complex result vector.
Available since version 3.

Properties:

MXX_TOP : double
value of MXX designing moment of top reinforcement
Available since version 3.

MXX_BOTTOM : double
value of MXX designing moment of bottom reinforcement
Available since version 3.

MYY_TOP : double
value of MYY designing moment of top reinforcement
Available since version 3.

MYY_BOTTOM : double
value of MYY designing moment of bottom reinforcement
Available since version 3.

S_MISES : double
reduced stress value for surface finite elements
Available since version 3.

N_MISES : double
value of reduced membrane force for surface finite elements

Available since version 3.

M_MISES : double
value of reduced moment for surface finite elements
Available since version 3.

IRobotFeResultReinforcement

Result vector for reinforcement.
Available since version 3.

Properties:

AY_TOP : double
Top reinforcement area (Y axis)
Available since version 3.

AY_BOTTOM : double
Bottom reinforcement area (Y axis)
Available since version 3.

AX_TOP : double
Top reinforcement area (X axis)
Available since version 3.

AX_BOTTOM : double
Bottom reinforcement area (X axis)
Available since version 3.

E_AY_TOP : double
Top reinforcement spacing (Y axis)
Available since version 3.

E_AY_BOTTOM : double
Bottom reinforcement spacing (Y axis)
Available since version 3.

E_AX_TOP : double
Top reinforcement spacing (X axis)
Available since version 3.

E_AX_BOTTOM : double
Bottom reinforcement spacing (X axis)
Available since version 3.

A_MIN : double
Minimum reinforcement area for one layer and one reinforcement direction
Available since version 3.

AY : double
Cracking (Y axis direction)
Available since version 3.

AX : double
Cracking (X axis direction)
Available since version 3.

F : double
deflection value
Available since version 3.

CalcError : bool

Available since version 3.

IRobotFeResultReduced

Reduced results for panels.
Available since version 3.

Properties:

NX : double
NRx reduced force in the panel plane
Available since version 3.

MZ : double
MRz reduced moment in the panel plane
Available since version 3.

TY : double
TRy reduced force in the panel plane
Available since version 3.

SO : double
sRo normal stress in the middle layer
Available since version 3.

SE : double
sRe normal stress in the middle layer
Available since version 3.

T : double
tR shear stress in the middle layer
Available since version 3.

TZ : double
TRz shear force outside the panel plane
Available since version 3.

MY : double
MRy reduced moment outside the panel plane

Available since version 3.

Length : double
Length of panel cut
Available since version 3.

Height : double
Panel height
Available since version 3.

CutPos : IRobotFeResultReducedCutPosition
definition of the panel cut
Available since version 5.5.

NodeLeftTop : long

Available since version 5.5.

NodeRightTop : long

Available since version 5.5.

NodeLeftBottom : long

Available since version 5.5.

NodeRightBottom : long

Available since version 5.5.

IRobotFeResultReducedCutPosition

Available positions of a panel cut.
Available since version 3.

Properties:

I_FRRCP_VERTICAL_LEFT : = 4

Available since version 3.

I_FRRCP_HORIZONTAL_MIDDLE : = 2

Available since version 3.

I_FRRCP_VERTICAL_RIGHT : = 6

Available since version 3.

I_FRRCP_HORIZONTAL_TOP : = 3

Available since version 3.

I_FRRCP_VERTICAL_MIDDLE : = 5

Available since version 3.

I_FRRCP_HORIZONTAL_BOTTOM : = 1

Available since version 3.

IRobotPanelCutMngr

Manager of panel cuts.
Available since version 3.

Properties:

Count : long
Number of available cuts
Available since version 3.

Operations:

Get (_idx : long) : IRobotPanelCut
Function returns the cut of the indicated index.
Available since version 3.

Create () : IRobotPanelCut
Function creates and returns a new cut.
Available since version 3.

Find (_cut_name : string) : long
Function returns index of the cut of the specified name.
Available since version 3.

Store (_cut : IRobotPanelCut, _name : string)
Function saves the cut under the specified name.
Available since version 3.

GetName (_idx : long) : string
Function returns name of the cut of the indicated index.
Available since version 3.

Remove (_idx : long)
Function deletes the cut of the indicated index.
Available since version 3.

IRobotPanelCut

Interface defining a cut through panel.
Available since version 3.

Properties:

Point1 : IRobotGeoPoint3D

Available since version 3.

Point2 : IRobotGeoPoint3D

Available since version 3.

Point3 : IRobotGeoPoint3D

Available since version 3.

Color : long
color applied while presenting a cut in graphical views
Available since version 3.

DefType : IRobotPanelCutDefinitionType
type of a panel cut definition
Available since version 3.

Active : bool
flag indicating if a given cut is active
Available since version 3.

IRobotPanelCutDefinitionType

Types of panel cut definition.
Available since version 3.

Properties:

I_PCDT_FULL_PLANE : = 1
unlimited cut plane, determined by two indicated points
Available since version 3.

I_PCDT_LIMITED_PLANE : = 2
a cut plane defined by a segment determined by two indicated points and perpendicular to XY plane
Available since version 3.

IRobotFeResultType

Designation of the type of result calculated for finite elements.
Available since version 4.

Properties:

I_FRT_DETAILED_SXX : = 483

Available since version 4.

I_FRT_DETAILED_SYY : = 484

Available since version 4.

I_FRT_DETAILED_SXY : = 485

Available since version 4.

I_FRT_DETAILED_NXX : = 492

Available since version 4.

I_FRT_DETAILED_NYY : = 493

Available since version 4.

I_FRT_DETAILED_NXY : = 494

Available since version 4.

I_FRT_DETAILED_MXX : = 501

Available since version 4.

I_FRT_DETAILED_MYY : = 502

Available since version 4.

I_FRT_DETAILED_MXY : = 503

Available since version 4.

I_FRT_DETAILED_TXX : = 510

Available since version 4.

I_FRT_DETAILED_TYY : = 511

Available since version 4.

I_FRT_DETAILED_QXX : = 519

Available since version 4.

I_FRT_DETAILED_QYY : = 520

Available since version 4.

I_FRT_DETAILED_UXX : = 537

Available since version 4.

I_FRT_DETAILED_UYY : = 538

Available since version 4.

I_FRT_DETAILED_WNORM : = 531

Available since version 4.

I_FRT_DETAILED_RXX : = 546

Available since version 4.

I_FRT_DETAILED_RYY : = 547

Available since version 4.

I_FRT_DETAILED_RNORM : = 549

Available since version 4.

I_FRT_DETAILED_PNORM : = 558

Available since version 4.

I_FRT_PRINCIPAL_S1 : = 487

Available since version 4.

I_FRT_PRINCIPAL_S2 : = 488

Available since version 4.

I_FRT_PRINCIPAL_S1_2 : = 490

Available since version 4.

I_FRT_PRINCIPAL_SAL : = 489

Available since version 4.

I_FRT_PRINCIPAL_N1 : = 496

Available since version 4.

I_FRT_PRINCIPAL_N2 : = 497

Available since version 4.

I_FRT_PRINCIPAL_N1_2 : = 499

Available since version 4.

I_FRT_PRINCIPAL_NAL : = 498

Available since version 4.

I_FRT_PRINCIPAL_M1 : = 505

Available since version 4.

I_FRT_PRINCIPAL_M2 : = 506

Available since version 4.

I_FRT_PRINCIPAL_M1_2 : = 508

Available since version 4.

I_FRT_PRINCIPAL_MAL : = 507

Available since version 4.

I_FRT_PRINCIPAL_T1_2 : = 517

Available since version 4.

I_FRT_PRINCIPAL_Q1_2 : = 526

Available since version 4.

I_FRT_PRINCIPAL_UGX : = 1302

Available since version 4.

I_FRT_PRINCIPAL_UGY : = 1303

Available since version 4.

I_FRT_PRINCIPAL_UGZ : = 1304

Available since version 4.

I_FRT_PRINCIPAL_U : = 1305

Available since version 4.

I_FRT_COMPLEX_S_MISES : = 491

Available since version 4.

I_FRT_COMPLEX_N_MISES : = 500

Available since version 4.

I_FRT_COMPLEX_M_MISES : = 509

Available since version 4.

I_FRT_COMPLEX_MXX_TOP_WA : = 889

Available since version 4.

I_FRT_COMPLEX_MXX_BOTTOM_WA : = 890

Available since version 4.

I_FRT_COMPLEX_MYY_TOP_WA : = 891

Available since version 4.

I_FRT_COMPLEX_MYY_BOTTOM_WA : = 892

Available since version 4.

I_FRT_COMPLEX_MXX_TOP_NEN : = 893

Available since version 4.

I_FRT_COMPLEX_MXX_BOTTOM_NEN : = 894

Available since version 4.

I_FRT_COMPLEX_MYY_TOP_NEN : = 895

Available since version 4.

I_FRT_COMPLEX_MYY_BOTTOM_NEN : = 896

Available since version 4.

I_FRT_REINF_AX_TOP : = 0

Available since version 4.

I_FRT_REINF_AX_BOTTOM : = 1

Available since version 4.

I_FRT_REINF_AY_TOP : = 2

Available since version 4.

I_FRT_REINF_AY_BOTTOM : = 3

Available since version 4.

I_FRT_REINF_E_AX_TOP : = 4

Available since version 4.

I_FRT_REINF_E_AX_BOTTOM : = 5

Available since version 4.

I_FRT_REINF_E_AY_TOP : = 6

Available since version 4.

I_FRT_REINF_E_AY_BOTTOM : = 7

Available since version 4.

I_FRT_REINF_A_MIN : = 8

Available since version 4.

I_FRT_REINF_AX : = 9

Available since version 4.

I_FRT_REINF_AY : = 10

Available since version 4.

I_FRT_REINF_F : = 11

Available since version 4.

I_FRT_REDUCED_NX : = 1217

Available since version 4.

I_FRT_REDUCED_MZ : = 1218

Available since version 4.

I_FRT_REDUCED_TY : = 1219

Available since version 4.

I_FRT_REDUCED_SO : = 1220

Available since version 4.

I_FRT_REDUCED_SE : = 1221

Available since version 4.

I_FRT_REDUCED_T : = 1222

Available since version 4.

I_FRT_REDUCED_TZ : = 1223

Available since version 4.

I_FRT_REDUCED_MY : = 1224

Available since version 4.

I_FRT_REDUCED_LENGTH : = 1225

Available since version 4.

I_FRT_REDUCED_HEIGHT : = 1226

Available since version 4.

IRobotFeResultSmoothing

Available methods of averaging (smoothing) results.
Available since version 4.

Properties:

I_FRS_NO_SMOOTHING : = 0
exact results in nodes
Available since version 4.

I_FRS_GLOBAL_SMOOTHING : = 1
globally averaged results in nodes
Available since version 4.

I_FRS_SMOOTHING_WITHIN_A_PANEL : = 2
results in nodes - averaged within a panel
Available since version 4.

I_FRS_SMOOTHING_ACCORDING_TO_SELECTION : = 3
results in nodes averaged according to the selection of finite elements to which the nodes belong
Available since version 4.

I_FRS_IN_ELEMENT_CENTER : = 4
averaged results in finite element centers
Available since version 4.

IRobotFeResultKind

Available since version 4.

Properties:

I_FRK_PANEL_NODE : = 0

Available since version 4.

I_FRK_ELEMENT_CENTER : = 1

Available since version 4.

I_FRK_ELEMENT_NODE : = 2

Available since version 4.

IRobotFeExtremeParams

Parameters of searching extreme results for finite elements.

The list of nodes to be considered while searching extreme values is determined based on the settings in the fields NodeSel, ElementSel and PanelSel.
If a list of nodes is set in the NodeSel field, then this list exactly will be considered in searching extreme values.
If the NodeSel field is empty and the ElementSel field is set, then all the nodes belonging to the indicated finite elements will be considered.
If the fields NodeSel and ElementSel are empty and the PanelSel field is set, then all the nodes belonging to the indicated panels will be considered.
If the fields NodeSel, PanelSel and ElementSel are empty, then all the structure nodes will be considered.

For results in finite element centers, the list of elements to be considered is determined based on the ElementSel field. If the ElementSel list is empty and the PanelSel field is set, then all the finite elements belonging to the indicated panels will be considered.
If the fields ElementSel and PanelSel are empty, then all the finite elements of the structure will be considered.

Available since version 4.

Properties:

ResultId : IRobotFeResultType
designation of the result type
Available since version 4.

PanelSel : string
list of panels to be considered
Available since version 4.

ElementSel : string

list of finite elements to be considered
Available since version 4.

NodeSel : string
list of nodes to be considered

Available since version 4.

CaseSel : string
list of load cases to be considered

Available since version 4.

Layer : IRobotFeLayerType
layer
Available since version 4.

LayerArbitraryValue : double
layer thickness
Available since version 4.

Smoothing : IRobotFeResultSmoothing
method of result averaging
Available since version 4.

ReducedCutPos : IRobotFeResultReducedCutPosition

Available since version 4.

CaseCmpnt : long
number of a load case component or mode number
Available since version 4.1.

ModeCmb : IRobotModeCombinationType
mode combination
Available since version 4.1.

Operations:

SetDirX (_def_type : IRobotObjLocalXDirDefinitionType, _x : double, _y : double, _z : double)

Available since version 4.

GetDirX (_x : double*, _y : double*, _z : double*) : IRobotObjLocalXDirDefinitionType

Available since version 4.

Reset ()

Available since version 4.

IRobotFeExtremeValue

Interface providing access to information about the extreme value.
Available since version 4.

Properties:

ResultId : IRobotFeResultType

Available since version 4.

Value : double

Available since version 4.

IsAvailable : bool

flag indicating if the extreme value has been taken correctly from the result server
Available since version 4.

Smoothing : IRobotFeResultSmoothing

Available since version 4.

Node : long

Available since version 4.

Panel : long

Available since version 4.

Element : long

Available since version 4.

Case : long

Available since version 4.

CaseCmpnt : long

additional number indicating a load case component
- for a code combination it is the number of its component
- for a modal analysis case it is the number of a mode
- for results of time history analysis it is the number of a step
Available since version 4.

ModeCmb : IRobotModeCombinationType

Available since version 4.

Layer : IRobotFeLayerType

Available since version 4.

LayerArbitraryValue : double

Available since version 4.

ReducedCutPos : IRobotFeResultReducedCutPosition

Available since version 4.

Operations:

GetDirX (_x : double*, _y : double*, _z : double*) : IRobotObjLocalXDirDefinitionType
Function takes the direction of the local X axis of the coordinate system, in which the specified extreme value has been obtained.
The values read are the coordinates of the X axis direction vector for definition in the Cartesian system or the coordinates of the central point for definition in the polar system. If values of all the arguments: x, y, z are set as zeros, then the system direction is compatible with the panel local system. In addition, the function returns the type of the system direction definition.
Available since version 4.

IRobotFeMultiResultType

Interface which enables indicating simultaneously - a great number of result types for finite elements.
Available since version 4.1.

Properties:

Count : long

Available since version 4.1.

Operations:

Add (_result_type : IRobotFeResultType)

Available since version 4.1.

Remove (_result_type : IRobotFeResultType)

Available since version 4.1.

Get (_idx : long) : IRobotFeResultType

Available since version 4.1.

IRobotFeMultiExtremeValue

Interface providing access to a great number of min./max. results for finite elements.
Available since version 4.1.

Properties:

Count : long

Available since version 4.1.

Operations:

GetByResType (_result_type : IRobotFeResultType) : IRobotFeExtremeValue
Function returns the indicated min./max. result.
Available since version 4.1.

Get (_idx : long) : IRobotFeExtremeValue

Available since version 4.1.

Available since version 7.5.

IRobotMaterialQuantitySurvey

Available since version 7.5.

Properties:

Count : long
number of all different materials used in a structure
Available since version 7.5.

Operations:

GetName (_idx : long) : string
Function returns a material name.
Available since version 7.5.

GetWeight (_obj_type : IRobotObjectType, _idx_or_name : VARIANT) : double
Function returns the total weight of all structure components of the specified type and of the specified material.
Available since version 7.5.

GetVolume (_obj_type : IRobotObjectType, _idx_or_name : VARIANT) : double
Function returns the total volume of all structure components of the specified type and of the specified material.
Available since version 7.5.

GetType (_idx_or_name : VARIANT) : IRobotMaterialType
Function returns a type of the given material.
Available since version 7.5.

IRobotBarSectionQuantitySurvey

Available since version 7.5.

Properties:

Count : long
number of different bar sections used in a structure
Available since version 7.5.

Operations:

GetName (_idx : long) : string
Function returns the name of a section of the given index
Available since version 7.5.

GetUnitWeight (_idx_or_name : VARIANT) : double
Function returns a unit weight of a selected section.
Available since version 7.5.

GetWeight (_idx_or_name : VARIANT) : double
Function returns the total weight of all structure members of a selected section.
Available since version 7.5.

GetVolume (_idx_or_name : VARIANT) : double
Function returns the total volume of all structure members of a selected section.
Available since version 7.5.

GetPaintArea (_idx_or_name : VARIANT) : double
Function returns the total painting area of all structure members of a selected section.
Available since version 7.5.

GetLength (_idx_or_name : VARIANT) : double
Function returns the total length of all structure members of a selected section.
Available since version 7.5.

IRobotThicknessQuantitySurvey

Available since version 7.5.

Properties:

Count : long
number of different panel thicknesses used in a structure
Available since version 7.5.

Operations:

GetName (_idx : long) : string
Function returns a thickness name of the given index.
Available since version 7.5.

GetUnitWeight (_idx_or_name : VARIANT) : double
Function returns a unit weight for a selected thickness.
Available since version 7.5.

GetWeight (_idx_or_name : VARIANT) : double
Function returns the total weight of all structure panels of a selected thickness.
Available since version 7.5.

GetVolume (_idx_or_name : VARIANT) : double
Function returns the total volume of all structure panels of a selected thickness.
Available since version 7.5.

GetArea (_idx_or_name : VARIANT) : double
Function returns the total area of all structure panels of a selected thickness.
Available since version 7.5.

IRobotStructureQuantitySurvey

Quantity survey of a structure.
Available since version 7.5.

Properties:

Materials : IRobotMaterialQuantitySurvey

Available since version 7.5.

BarSections : IRobotBarSectionQuantitySurvey

Available since version 7.5.

PanelThickness : IRobotThicknessQuantitySurvey

Available since version 7.5.

Edit operations

Available since version 3.5.

IRobotTranslateOptions

Available since version 3.5.

Properties:

I_TO_COPY : = 0

Available since version 3.5.

I_TO_COPY_WITH_DRAG : = 1

Available since version 3.5.

I_TO_MOVE : = 2

Available since version 3.5.

IRobotStructureEditTools

Interface grouping edit functions that operate on the entire structure or its part.
Available since version 3.5.

Operations:

DivideBar (_bar_number : long, _div_points : IRobotValuesArray, _relative : bool) : IRobotNumbersArray
Function divides the indicated bar based on the provided table with positions of division points.
Function returns tables with numbers of bars generated as a result of the division operation.
Available since version 3.5.

TranslateBar (_bar_number : long, _dx : double, _dy : double, _dz : double, _trans_option : IRobotTranslateOptions, _repetitions : long = 1) : IRobotNumbersArray
Function translates selected bar by the specified vector.
Function returns a table with numbers of bars generated as a result of translation.
Available since version 3.5.

TranslateNode (_node_number : long, _dx : double, _dy : double, _dz : double, _trans_option : IRobotTranslateOptions, _repetitions : long) : IRobotNumbersArray
Function translates the selected node by the specified vector.
Function returns a table with numbers of nodes generated as a result of translation.

Available since version 3.5.

SelTranslate (_dx : double, _dy : double, _dz : double, _options : IRobotTranslateOptions, _repetitions : long)
Function performs the operation of translation on the current selection.
Available since version 4.

SelRotate (_p0x : double, _p0y : double, _p0z : double, _p1x : double, _p1y : double, _p1z : double, _angle : double, _options : IRobotTranslateOptions, _repetitions : long)
Function performs the operation of rotation on the current selection.
Available since version 4.

SelScale (_p0x : double, _p0y : double, _p0z : double, _factor : double, _options : IRobotTranslateOptions, _repetitions : long)
Function performs the operation of scaling on the current selection.
Available since version 4.

SelMirror (_p0x : double, _p0y : double, _p0z : double, _p1x : double, _p1y : double, _p1z : double, _p2x : double, _p2y : double, _p2z : double, _options : IRobotTranslateOptions, _repetitions : long)

Available since version 4.

Storeys

IRobotStorey

Interface describing a story in a structure.

Properties:

Name : string
story name

Level : double
start level (floor) of a story

Height : double
story height

Color : long
color assigned to the story

AutomaticSelection : bool
flag indicating if selection of objects assigned to a story should be automatic

Objects : string
selection of objects assigned to a story

IRobotStoreyMngr

Structure story manager.

Properties:

DisregardedObjects : string
selection of objects that should be disregarded in automatic recognition of structure elements belonging to individual stories

Count : long
number of stories defined in a structure

Operations:

Get (_idx : long) : IRobotStorey
Function returns definition of a story with the specified index. Stories are indexed with numbers from 1 to Count.

Find (_story_name : string) : long
Function searches for the story with the specified name and returns its index. If the story with the given name does not exist, function returns zero value.

Delete (_idx : long)
Function deletes the story with the given index.

DeleteAll ()
Function deletes all stories defined in a structure.

CreateEx (_name : string, _level : double, _height : double, _objects : string = "", _color : long = -1) : long
Function creates a story with the specified parameters. Function returns the index of the created story.

Create (_name : string, _level : double, _height : double) : long
Function creates a new story with the specified parameters and returns its index. The created story is assigned by default an automatic color and an automatic object selection.

Calculation module

IRobotCalcEngine

Interface representing the calculation module of Robot program.

Properties:

AutoGenerateModel : bool
flag steering automatic model generation that directly precedes structure calculations
Available since version 2.5.

AnalysisParams : IRobotStructureAnalysisParams
Parameters of structure analysis
Available since version 3.

AutoFreezeResults : bool
flag enforcing automatic freezing of results after calculations
Available since version 3.

SaveResultsInExternalFile : bool
flag enforcing result saving in the external file *.RT_
Available since version 3.

SeismicResultsSaveParams : IRobotSeismicResultsSaveParams
saving parameters for seismic analysis results
Available since version 3.

GenerationParams : IRobotModelGenerationParams
Parameters of calculation model generation
Available since version 3.

BucklingDeformation : IRobotBucklingDeformationParams
Buckling deformation
Available since version 3.

UseStatusWindow : bool
flag that steers the status window display while performing time-consuming calculations
Available since version 5.5.

StatusWindowParent : long
handle to parent window for the calculation status window
Available since version 5.5.

Operations:

Calculate () : bool
The function recalculates the entire structure. If there appeared an error during calculations, the function returns zero value (False), otherwise it returns a non-zero value (True)

GenerateModel ()
Function generates a structure calculation model.
Available since version 3.5.

IRobotEquationSolvingMethod

Available methods of equation set solution.
Available since version 3.

Properties:

I_ESM_FRONTAL : = 0

Available since version 3.

I_ESM_SKYLINE : = 1

Available since version 3.

I_ESM_SPARSE : = 4

Available since version 3.

I_ESM_SPARSE_M : = 5

Available since version 3.

I_ESM_ITERATIVE : = 2

Available since version 3.

I_ESM_AUTO : = 3

Available since version 3.

IRobotStructureAnalysisParams

Parameters of structure analysis
Available since version 3.

Properties:

EquationSolvingMethod : IRobotEquationSolvingMethod
selected method of equation set solution

Available since version 3.

AutoVerification : IRobotStructureAutoVerificationType
selected type of automatic structure verification
Available since version 3.

IgnoreWarnings : bool

Available since version 3.

DSCAlgorithm : bool
DSC algorithm
Available since version 3.

FictitiousRigidityCoeff : double
Fictitious rigidity coefficient
Available since version 3.

SparseMParams : IRobotSparseMSolverParams

Available since version 3.

IterativeParams : IRobotIterativeSolverParams

Available since version 3.

ModalParticipationCoeff : IRobotStructureAnalysisModalParticipationCoeff
method of participation coefficient calculation
Available since version 3.

AutoBarMerging : bool
Automatic bar merging during import of geometry
Available since version 3.

IRobotStructureAutoVerificationType

Available types of automatic structure verification.
Available since version 3.

Properties:

I_SAVT_NONE : = 0

Available since version 3.

I_SAVT_ERRORS_ONLY : = 1

Available since version 3.

I_SAVT_ERRORS_AND_WARNINGS : = 2

Available since version 3.

IRobotSparseMSolverParams

Available since version 3.

Properties:

Method : IRobotSparseMSolverMethod

Available since version 3.

IRobotSparseMSolverMethod

Available since version 3.

Properties:

I_SMSM_NDM : = 3
Nested dissection method
Available since version 3.

I_SMSM_MDA : = 4
Minimal degrees algorithm (MDA)
Available since version 3.

IRobotIterativePrediconditionerType

Available since version 3.

Properties:

I_IPT_DIAGONAL : = 0

Available since version 3.

I_IPT_GAUSS : = 1

Available since version 3.

I_IPT_CHOLESKY : = 2

Available since version 3.

I_IPT_ICCF : = 3

Available since version 3.

IRobotIterativeSolverParams

Available since version 3.

Properties:

PrediconditionerType : IRobotIterativePrediconditionerType

Available since version 3.

Multilevel : bool

Available since version 3.

AggregationLevelsCount : long
Number of aggregation levels
Available since version 3.

InternalIterationsCount : long
Number of internal iterations
Available since version 3.

AnalyseDiagonale : bool
Diagonal analysis
Available since version 3.

CalcKMatrix : bool
calculation of the K matrix
Available since version 3.

Method : IRobotIterativeSolverMethod

Available since version 3.

MemoryUsage : IRobotIterativeSolverMemoryUsage

Available since version 3.

Tolerance : double
Tolerance definition
Available since version 3.

IRobotIterativeSolverMethod

Available since version 3.

Properties:

I_ISM_0 : = 0

Available since version 3.

I_ISM_1 : = 1

Available since version 3.

I_ISM_2 : = 2

Available since version 3.

IRobotIterativeSolverMemoryUsage

Available since version 3.

Properties:

I_ISMU_MIN : = 0

Available since version 3.

I_ISMU_1_2 : = 2

Available since version 3.

I_ISMU_1_4 : = 1

Available since version 3.

I_ISMU_MAX : = 3

Available since version 3.

IRobotStructureAnalysisModalParticipationCoeff

Available methods of participation coefficient calculation.
Available since version 3.

Properties:

I_SAMPC_SUM_ABSOLUTE_VALUES : = 0
Sum of absolute values
Available since version 3.

I_SAMPC_SQUARE_ROOT_OF_SUM_SQUARES : = 1
Square root of sum of squares
Available since version 3.

IRobotSeismicResultsSaveParams

Parameters of saving quadratic combination results for seismic analysis in a project.
Available since version 3.

Properties:

Save : bool
flag indicating if results are saved in the project
Available since version 3.

Reactions : bool
flag enforcing saving of reactions
Available since version 3.

Displacements : bool
flag enforcing saving of displacements

Available since version 3.

Forces : bool
flag enforcing saving of forces
Available since version 3.

Stresses : bool
flag enforcing saving of stresses
Available since version 3.

NMQ : bool
flag enforcing saving of N membrane forces, M moments and Q shear forces
Available since version 3.

Reduced : bool
flag enforcing saving of reduced results
Available since version 3.

PointNumber : long
definition of a number of points along the bar length
Available since version 3.

PanelsDir : IRobotSeismicResultsPanelDirection
Direction for panels
Available since version 3.

OnlyQuadraticCombs : bool
result save only for quadratic combinations
Available since version 5.6.

LocalDisplacements : bool

local displacements

IRobotSeismicResultsPanelDirection

Available since version 3.

Properties:

I_SRPD_AUTOMATIC : = 0

Available since version 3.

I_SRPD_DIR_X : = 1

Available since version 3.

I_SRPD_DIR_Y : = 2

Available since version 3.

I_SRPD_DIR_Z : = 3

Available since version 3.

IRobotModelGenerationParams

Generation parameters for structure calculation model.
Available since version 3.

Properties:

NeglectedBars : string
list of bars to be disregarded during generation of the calculation model
Available since version 3.

NeglectedGeoObjects : string
list of geometrical objects to be disregarded during generation of the calculation model
Available since version 3.

AssemblingCase : long
assembling case number
Available since version 3.

ToleranceAutomatic : bool
enforcing the automatic value for tolerance
Available since version 3.

ToleranceValue : double
Tolerance value
Available since version 3.

GenerateNodes_DiagonalBars : bool
node generation at intersections of inclined bars
Available since version 3.

GenerateNodes_VertHorizBars : bool
node generation at intersections of vertical and horizontal bars
Available since version 3.

GenerateNodes_BarsAndFiniteElems : bool
node generation at intersections of bars and finite elements
Available since version 3.

MaxElementLength : double
Maximum length of calculation element
Available since version 3.

IRobotBucklingDeformationParams

Parameters of buckling deformation
Available since version 3.

Properties:

CaseNumber : long
number of selected buckling analysis case (zero value indicates that no case has been selected)
Available since version 3.

MaxDisplacement : double
Maximum displacement
Available since version 3.

OmitCaseForDeformations : bool
flag indicating if a selected buckling case is to be disregarded for a structure including deformations
Available since version 3.

Operations:

SetModeCoeff (_mode : long, _coeff : double)
Function defines coefficient for the indicated mode.
Available since version 3.

GetModeCoeff (_mode : long) : double
Function returns coefficient for the indicated mode.
Available since version 3.

Data presentation

Structural axes

IRobotStructuralAxisLabelType

Available methods of structure axis description.

Properties:

I_SALT_123 : = 1
description with successive numbers

I_SALT_ABC : = 2
description with successive alphabet letters

I_SALT_VALUE : = 3
description with values

I_SALT_DEFINE : = 4
description with a defined character string

I_SALT_VARIOUS : = 5
Application of different methods of axis description

IRobotStructuralAxisSequenceList

List of structure axis sequences.

Properties:

SequenceCount : long
number of defined structure axis sequences

AxisCount : long
number of defined structure axes

StartPosition : double
beginning point of the list

Operations:

AddSequence (_distance : double, _repeat_number : long)
Function inserts a new structure axis sequence. While creating the first sequence, the start axis will also be created at the position StartPosition.

GetSequence (_idx : long, _distance : double*, _repeat_number : long*)
Function returns data for the sequence of a given index.

GetAxis (_idx : long, _label : string*, _position : double*, _singleout : bool*) : bool
Function returns data for the structure axis of a given index.

SingleOutAxis (_idx : long, _singleout : bool = true)
Function steers with singling out of a structure axis in graphical views.

SetAxisLabel (_idx : long, _label : string)
Function changes a description for the structure axis of a given index.

FindAxisByPos (_pos : double) : long
Function returns an index of the axis of a given position. If no structure axis has been defined at the given position, then function returns zero value.

DeleteSequence (_idx : long)
Function deletes the structure axis sequence of a given index.

Clear ()
Function deletes all structure axes.

SetLabelFormat (_label_type : IRobotStructuralAxisLabelType, _label_def : string = "", _reversed : bool = false)
Function sets the current method of structure axis description.

IRobotStructuralAxisGridType

Available types of structure axis grids.

Properties:

I_SAGT_CARTESIAN : = 1
Cartesian grid

I_SAGT_CYLINDRICAL : = 2
cylindrical grid

I_SAGT_ARBITRARY : = 3

IRobotStructuralAxisGrid

Structure axis grid.

Properties:

Name : string
grid name

Type : IRobotStructuralAxisGridType
grid type

Operations:

Save ()
Function saves the grid definition.

IRobotStructuralAxisGridCartesian

Derived from IRobotStructuralAxisGrid
Cartesian grid of structure axes.

Properties:

X : IRobotStructuralAxisSequenceList
sequence of structure axes defined on the OX axis

Y : IRobotStructuralAxisSequenceList
sequence of structure axes defined on the OY axis

Z : IRobotStructuralAxisSequenceList
sequence of structure axes defined on the OZ axis

RotationAxis : IRobotGeoCoordinateAxis
axis of grid rotation

RotationAngle : double
angle of grid rotation

IRobotStructuralAxisGridMngr

Manager of structure axis grids.

Properties:

Count : long
number of defined grids

Operations:

Create (_type : IRobotStructuralAxisGridType, _name : string) : IRobotStructuralAxisGrid
Function creates and returns a new blank grid of a defined type.

FindByName (_name : string) : long
Function returns an index of the grid of a given name or a zero value if the grid is not found.

Activate (_idx : long, _activate : bool = true)
Function activates/deactivates the grid of a given index.

Delete (_idx : long)
Function deletes the grid of a given index.

Clear ()
Function deletes all grids.

Get (_idx : long) : IRobotStructuralAxisGrid
Function makes available the grid of a given index.

GetByName (_name : string) : IRobotStructuralAxisGrid
Function makes available the grid of a given name.

IsActive (_idx : long) : bool
Function returns activity state of the grid of a given index.

Views and layouts

IRobotViewMngr

View manager controls all the views and layouts available in Robot.

Properties:

CurrentLayout : IRobotLayoutId
Identifier of active layout
Available since version 3.

ViewCount : long
number of opened graphical views
Available since version 3.

TableCount : long
number of opened tables
Available since version 3.

RecycleViews : bool

Operations:

CreateTable (tab_type : IRobotTableType, data_type : IRobotTableDataType) : IRobotTable
The function creates and returns a table of the indicated type that will present the appropriate type of data.

CreateView (view_type : IRobotViewType) : IRobotView
The function creates and returns the graphical view of the indicated type.

Refresh ()
The function results in refreshing all the displayed views.

CreateRtfView () : IRobotRtfView
The function creates a view for files in RTF format.

CreateFromSc (sc_name : string) : IDispatch
Function creates and returns a view on the basis of a screen capture with the specified name. Type of the created view depends on the screen capture contents. If structure view manager is unable to make any of the defined view interfaces available, then, IRobotPrintable interface is returned. If a screen capture with the specified name does not exist or if during its reading fro the archive an error occurs, then, empty reference will be returned (zero from the level of C language, Nothing from the level of Visual Basic).
Available since version 1.7.

GetView (_view_idx : long) : IRobotView2
Function returns the interaface for the graphical view of the indicated index.
Available since version 3.

GetTable (_table_idx : long) : IRobotTableFrame
Function returns the interface for the table of the specified index.
Available since version 3.

IRobotLayoutId

Set of identifiers for the layouts defined in the Robot program.
Available since version 3.

Properties:

I_LI_MODEL_GEOMETRY : = 1

Available since version 3.

I_LI_MODEL_NODES : = 2

Available since version 3.

I_LI_MODEL_BARS : = 3

Available since version 3.

I_LI_MODEL_SUPPORTS : = 4

Available since version 3.

I_LI_MODEL_PROPERTIES : = 5

Available since version 3.

I_LI_MODEL_LOADS : = 6

Available since version 3.

I_LI_RC_COLUMN_DEFINITION : = 11

Available since version 3.

I_LI_RC_COLUMN_RESULTS : = 12

Available since version 3.

I_LI_RC_COLUMN_REINFORCEMENT : = 13

Available since version 3.

I_LI_RC_COLUMN_DRAWINGS : = 14

Available since version 3.

I_LI_RC_BEAM_DEFINITION : = 21

Available since version 3.

I_LI_RC_BEAM_SECTIONS : = 22

Available since version 3.

I_LI_RC_BEAM_HOLES : = 23

Available since version 3.

I_LI_RC_BEAM_LOADS : = 24

Available since version 3.

I_LI_RC_BEAM_RESULTS : = 25

Available since version 3.

I_LI_RC_BEAM_REINFORCEMENT : = 26

Available since version 3.

I_LI_RC_BEAM_WALL_DEFINITION : = 31

Available since version 3.

I_LI_RC_BEAM_WALL_SECTIONS : = 32

Available since version 3.

I_LI_RC_BEAM_WALL_HOLES : = 33

Available since version 3.

I_LI_RC_BEAM_WALL_LOADS : = 34

Available since version 3.

I_LI_RC_BEAM_WALL_RESULTS : = 35

Available since version 3.

I_LI_RC_BEAM_WALL_REINFORCEMENT : = 36

Available since version 3.

I_LI_FOOTING_DEFINITION : = 41

Available since version 3.

I_LI_FOOTING_LOADS : = 42

Available since version 3.

I_LI_FOOTING_SOIL : = 43

Available since version 3.

I_LI_FOOTING_RESULTS : = 44

Available since version 3.

I_LI_FOOTING_REINFORCEMENT : = 45

Available since version 3.

I_LI_DESIGN_CONNECTIONS : = 51

Available since version 3.

I_LI_DESIGN_STEEL_ALUMINUM : = 52

Available since version 3.

I_LI_DESIGN_WOOD : = 53

Available since version 3.

I_LI_DESIGN_RC_MEMBERS : = 54

Available since version 3.

I_LI_TOOL_SECTION_DEFINITION : = 61

Available since version 3.

I_LI_TOOL_TEXT_FILE : = 62

Available since version 3.

I_LI_TOOL_FINAL_DRAWING : = 63

Available since version 3.

I_LI_FOUNDATION_DEFINITION : = 71

Available since version 3.

I_LI_FOUNDATION_LOADS : = 72

Available since version 3.

I_LI_FOUNDATION_SOIL : = 73

Available since version 3.

I_LI_FOUNDATION_RESULTS : = 74

Available since version 3.

I_LI_FOUNDATION_REINFORCEMENT : = 75

Available since version 3.

I_LI_RESULTS_DIAGRAMS : = 81

Available since version 3.

I_LI_RESULTS_DETAILED : = 82

Available since version 3.

I_LI_RESULTS_STRESS_BARS : = 83

Available since version 3.

I_LI_RESULTS_STRESS_STRUCTURE : = 84

Available since version 3.

I_LI_RESULTS_MAPS : = 85

Available since version 3.

I_LI_RC_SLAB_REQUIRED_REINF : = 91

Available since version 3.

I_LI_RC_SLAB_PROVIDED_REINF : = 92

Available since version 3.

I_LI_RC_SLAB_REINFORCEMENT : = 93

Available since version 3.

Graphical structure views

IRobotViewType

Set of available types of graphical views of the structure that can be created by means of the view manager.

Properties:

I_VT_STANDARD : = 1
standard structure view

I_VT_DETAILED_ANALYSIS : = 2
a view presenting detailed analysis results
Available since version 1.7.

I_VT_DIAGRAMS : = 3
Available since version 2.0.

I_VT_MAPS_ON_BARS : = 4
Available since version 2.0.

I_VT_MAPS_ON_FINITE_ELEMENTS : = 5
Available since version 2.0.

I_VT_PANEL_CUTS : = 6

Available since version 3.

I_VT_GLOBAL_ANALYSIS : = 7

Available since version 3.

I_VT_INFLUENCE_LINES : = 8

Available since version 3.5.

IRobotView

Interface representing the graphical structure view.

Properties:

Type : IRobotViewType
view type

Visible : bool
view display flag

Title : string
view title

Printable : IRobotPrintable
object representing the view that may be attached to the printout and printed

Selection : IRobotMultiSelection
selection of structure components presented in the view

Projection : IRobotViewProjection
current projection

SymbolMagnification : int
symbol enlarging (scale: 1 to 10)

DiagramMagnification : int
diagram enlarging (scale: 1 to 10)

GridVisible : bool

visibility status of a grid

RulerVisible : bool
visibility status of a ruler

StructAxisVisible : bool
visibility status of structure axes

ParamsDisplay : IRobotViewDisplayParams
display parameters (a component enabling control of individual attribute display)
Available since version 2.0.

ParamsDiagram : IRobotViewDiagramParams
parameters of displaying diagrams on the view
Available since version 2.0.

ParamsBarMap : IRobotViewBarMapParams
parameters of displaying maps an bars
Available since version 2.0.

ParamsFeMap : IRobotViewFeMapParams
display parameters of maps presenting results for surface finite elements
Available since version 2.0.

Operations:

Redraw (force_init_zoom : bool)
The function draws the view.

GetSize (width_in_mm : double*, height_in_mm : double*)
The function returns the size of the view in mm.

SetSize (width_in_mm : double, height_in_mm : double)
The function sets the view size in mm

GetZoom (left : double*, top : double*, right : double*, bottom : double*)
The function returns the current zoom - coordinates of view sides in the current coordinate system

SetZoom (left : double, top : double, right : double, bottom : double)
The function sets the current zoom - coordinates of view sides in the current coordinate system

GetVisibilityStatus (status_id : IRobotViewVisibilityStatusType) :
IRobotViewVisibilityStatusValue

The function gives back the value of the given visibility status.

SetVisibilityStatus (status_id : IRobotViewVisibilityStatusType, status_value : IRobotViewVisibilityStatusValue)
The function sets a value of the given visibility status.

GetScale (scale_id : int) : double
The function gives a value of the given scale back. A set of available scales describes RobotViewScaleType type.

SetScale (scale_id : int, val : double)
The function assigns a given value to the scale.

IsLocal () : bool
The function gives the status of a local coordinate system back.

SetGlobal ()
The function sets a global coordinate system.

SetLocal (p1 : IRobotGeoPoint3D, p2 : IRobotGeoPoint3D, p3 : IRobotGeoPoint3D) : bool
The function sets a local coordinate system. If the given points do not describe a local plane in the unambiguous way the function gives a zero value (False) back.

GetRotationPoint () : IRobotGeoPoint3D
The function gives back a rotation center during a view rotation.

SetRotationPoint (rot_point : IRobotGeoPoint3D)
The funcion sets the rotation center during the view rotation.

GetWorkPoint () : IRobotGeoPoint3D
The function gives a work point back (on the XY plane it gives Z level)

SetWorkPoint (point : IRobotGeoPoint3D)
The function sets a working point (for the XY plane Z level is set).

CopyToClipboard ()
The function inserts the view into the clipboard.

Rotate (axis : IRobotGeoCoordinateAxis, angle : double)
The function rotates structure view with respect to the indicated global coordinate axis, with the rotation point recognized.

IRobotViewProjection

The set of identifiers determining different types of projections to be presented in graphical views has been defined.

Properties:

I_VP_XY : = 0
I_VP_XZ : = 1
I_VP_YZ : = 2
I_VP_3DXYZ : = 3
I_VP_XY_3D : = 4
deep projection (XY plane)

I_VP_YZ_3D : = 5
deep projection (YZ plane)

I_VP_XZ_3D : = 6
deep projection (XZ plane)

IRobotViewVisibilityStatusType

Available types of visibility status for a graphical viewer.

Properties:

I_VVST_STRUCTURE_3D : = 0
3D structure

I_VVST_SECTIONS : = 1
sections

I_VVST_ANIMATION : = 2
animation

I_VVST_SUPERPOS : = 3
case superposition

I_VVST_MODE : = 4
modes

I_VVST_BUCKLING : = 5
buckling

I_VVST_PLOT : = 6
a plotter drawing

I_VVST_SCALE : = 7
scale

I_VVST_FILL : = 8
a method of area filling

I_VVST_ACTIVE_CASE : = 9
a number of active case

I_VVST_MOBILE : = 10
settings for moving loads

I_VVST_ACTIVE_MODE : = 11
a number of active mode

I_VVST_ACTIVE_PHASE : = 12
an active structure phase

I_VVST_ACTIVE_QCMB : = 13
square combination mode

I_VVST_STRUCTURE : = 14
structure

I_VVST_NODES : = 15
node numbers

I_VVST_ELE : = 16
element numbers

I_VVST_NOT_SELECTED : = 17
objects not selected

I_VVST_EXPLODE : = 18
explode

I_VVST_MTC : = 19
materials, compression, tension

I_VVST_SECTION_NAMES : = 20
section names

I_VVST_SECTION_SYMB : = 21
section symbols

I_VVST_SECTION_DRAW : = 22
section drawings

I_VVST_SECTION_SURF : = 23
section area

I_VVST_LOCAL : = 24
local systems

I_VVST_THICKNESS : = 25
thicknesses

I_VVST_SUPPORTS : = 26
supports

I_VVST_RELEASES : = 27
releases

I_VVST_RIGID_LINKS : = 28
rigid links

I_VVST_LOADS : = 29
loads

I_VVST_NODE_LOADS : = 30
nodal loads

I_VVST_BAR_LOADS : = 31
bar loads

I_VVST_DEFORMATION : = 32
deformed structure

I_VVST_FORCES : = 33
forces

I_VVST_EXTREMES : = 34
extreme forces

I_VVST_Dia3D : = 35
I_VVST_DetDi : = 36
I_VVST_GloDi : = 37
I_VVST_Dep : = 38
I_VVST_CtrT : = 39
I_VVST_BEST : = 40
I_VVST_REACTIONS : = 41
residual reactions and vectors

I_VVST_FE : = 42
panels and finite elements

I_VVST_STRESSES : = 43
stress curvatures

I_VVST_FE_CUTS : = 44
cuts through finite elements

I_VVST_FE_DIRECTION : = 45
a direction of result calculations in finite elements

I_VVST_FE_LAY : = 46
a layer of stress calculations in finite elements

I_VVST_REINF_DIR : = 47
main direction of reinforcement

I_VVST_BAR_MAPS : = 48
maps on bars

I_VVST_TITLE : = 49
title

I_VVST_OFFSETS : = 50
offsets

I_VVST_COMPATIBILITIES : = 51
compatibilities

I_VVST_ENVELOPES : = 52
envelopes

I_VVST_ANALYSIS : = 53
analysis

I_VVST_DIM_LINES : = 54
dimension lines

I_VVST_REINF_CROSSES : = 55
reinforcement crosses

I_VVST_STRESSES_GLOBAL : = 56
Available since version 1.7.

I_VVST_DESCRIPTIONS : = 57
Available since version 2.0.

I_VVST_GRID : = 58
Available since version 2.0.

I_VVST_RULER : = 59
Available since version 2.0.

_VVST_STRUCTURAL_AXIS : = 60
Available since version 2.0.

I_VVST_MAPS : = 61
Available since version 2.0.

I_VVST_MAPS_LAYER : = 62
Available since version 2.0.

I_VVST_MAPS_DEFORMATION : = 63
Available since version 2.0.

IRobotViewVisibilityStatusValue

Accessible values for particular types of visibility status.

Properties:

I_VVSV_MTC_MATERIAL : = 0x01
material

I_VVSV_MTC_COMPRESSION : = 0x02
compression

I_VVSV_MTC_TENSION : = 0x04
tension

I_VVSV_MTC_CABLE : = 0x08
cables

I_VVSV_MTC_FEM : = 0x10
I_VVSV_STRUCTURE_ON : = 0x01
structure geometry

I_VVSV_STRUCTURE_POI : = 0x02
I_VVSV_STRUCTURE_HDL : = 0x04
hidden lines

I_VVSV_STRUCTURE_COR : = 0x08
I_VVSV_STRUCTURE_SHD : = 0x10
shading

I_VVSV_STRUCTURE_SPD : = 0x20
I_VVSV_STRUCTURE_HDF : = 0x40
colorful sides, black edges

I_VVSV_STRUCTURE_HDR : = 0x80
Shading with edge correction.

I_VVSV_STRUCTURE_HID : = (I_VVSV_STRUCTURE_HDL | I_VVSV_STRUCTURE_SHD | I_VVSV_STRUCTURE_HDR | I_VVSV_STRUCTURE_HDF)
I_VVSV_SUPPORTS_SYMB : = 0x01
support symbols

I_VVSV_SUPPORTS_COD : = 0x02
support codes

I_VVSV_SUPPORTS_LBL : = 0x04
labels for all visible attributes (instead of codes)

I_VVSV_ANALYSIS_DET : = 0x01
detailed analysis

I_VVSV_ANALYSIS_GLO : = 0x02
global analysis

I_VVSV_ANALYSIS_INF : = 0x08
analysis of influence lines

I_VVSV_ENVELOPES_MIN : = 0x01
minimum envelope

I_VVSV_ENVELOPES_MAX : = 0x02
maximum envelope

I_VVSV_ENVELOPES_CMP : = 0x04
all envelope components

I_VVSV_ENVELOPES_ONE : = 0x08
single components in complex cases

I_VVSV_ENVELOPES_HST : = 0x10
time history analysis case

I_VVSV_ENVELOPES_CUT : = 0x20
the analysis in cuts

I_VVSV_ENVELOPES_SEU : = 0x40
the analysis of a single bar

I_VVSV_ENVELOPES_SET : = 0x80
I_VVSV_SECTION_DRAW_SHAPE : = 0x01
section shapes

I_VVSV_SECTION_DRAW_COLOR : = 0x10
section colors

I_VVSV_SECTION_DRAW_TOP_BOTTOM : = 0x20
top and bottom of elements in colors

I_VVSV_SECTION_DRAW_GROUP : = 0x40
group colors

I_VVSV_SECTION_SYMB_SYMB : = 0x1
section symbols

I_VVSV_SECTION_SYMB_ELA : = 0x2
elastic ground for bars

I_VVSV_FILL_FILL : = 0x1
map filling

I_VVSV_FILL_AVG : = 0x2

averaging values in finite elements

I_VVSV_FILL_VAL : = 0x4
values on maps

I_VVSV_FILL_ISO : = 0x8
isolines

I_VVSV_FILL_BOR : = 0x10
contour

I_VVSV_FILL_TAN : = 0x20
cuts on normal plane

I_VVSV_FILL_DEF : = 0x40
deformation

I_VVSV_FILL_SMOOTH_NON : = 0
no smoothing

I_VVSV_FILL_SMOOTH_GLO : = 0x100
global smoothing

I_VVSV_FILL_SMOOTH_LOC : = 0x200
smoothing within panels

I_VVSV_FILL_SMOOTH_SEL : = 0x300
smoothing on selection

I_VVSV_RIGID_LINKS_RIG : = 0x1
rigid links

I_VVSV_RIGID_LINKS_SYM : = 0x2
rigid links - symbols

I_VVSV_RIGID_LINKS_DET : = 0x4
rigid links - details

I_VVSV_RIGID_LINKS_SUR : = 0x8
rigid links - areas

I_VVSV_FE_CON : = 0x1
contours

I_VVSV_FE_COI : = 0x2
inside part of contours

I_VVSV_FE_COL : = 0x4
thickness colors

I_VVSV_FE_COE : = 0x8

contour components

I_VVSV_FE_MES : = 0x10
mesh

I_VVSV_FE_EMI : = 0x20
emitters

I_VVSV_FE_RES : = 0x40
results

I_VVSV_FE_DSC : = 0x80
description of panels

I_VVSV_FE_DSO : = 0x100
description of objects

I_VVSV_FE_EDG : = 0x200
description of edges

I_VVSV_FE_ADV : = 0x400
detailed description of objects

I_VVSV_ELE_BAR_NUM : = 0x1
bar numbers

I_VVSV_ELE_CAL_NUM : = 0x2
calculation element numbers

I_VVSV_ELE_FE_NUM : = 0x4
finite element numbers

I_VVSV_ELE_FE_VIS : = 0x8
finite elements

I_VVSV_MOBILE_ROUTE : = 0x1
route

I_VVSV_MOBILE_CAR : = 0x2
vehicle

I_VVSV_MOBILE_ELE : = 0x4
elements

I_VVSV_MOBILE_LOADS : = 0x8
loads

I_VVSV_STRUCTURE_HDY : = 0x100
quick shading of faces
Available since version 1.7.

I_VVSV_FILL_SMOOTH_CAR : = 0x400
Available since version 1.7.

I_VVSV_FORCES_FX : = 0x0001
Available since version 1.7.

I_VVSV_FORCES_FY : = 0x0002
Available since version 1.7.

I_VVSV_FORCES_FZ : = 0x0004
Available since version 1.7.

I_VVSV_FORCES_FXC : = 0x0008
Available since version 1.7.

I_VVSV_FORCES_MX : = 0x0010
Available since version 1.7.

I_VVSV_FORCES_MY : = 0x0020
Available since version 1.7.

I_VVSV_FORCES_MZ : = 0x0040
Available since version 1.7.

I_VVSV_FORCES_UX : = 0x0100
Available since version 1.7.

I_VVSV_FORCES_UY : = 0x0200
Available since version 1.7.

I_VVSV_FORCES_UZ : = 0x0400
Available since version 1.7.

I_VVSV_FORCES_DFL : = 0x1000
Available since version 1.7.

I_VVSV_STRESSES_NORMALMIN : = 0x0001
Available since version 1.7.

I_VVSV_STRESSES_NORMALMAX : = 0x0002
Available since version 1.7.

I_VVSV_STRESSES_FLEXMIN_MY : = 0x0004
Available since version 1.7.

I_VVSV_STRESSES_FLEXMIN_MZ : = 0x0008
Available since version 1.7.

I_VVSV_STRESSES_FLEXMAX_MY : = 0x0010
Available since version 1.7.

I_VVSV_STRESSES_FLEXMAX_MZ : = 0x0020
Available since version 1.7.

I_VVSV_STRESSES_AXIAL : = 0x0040
Available since version 1.7.

I_VVSV_STRESSES_SHEAR_Y : = 0x0080
Available since version 1.7.

I_VVSV_STRESSES_SHEAR_Z : = 0x0100
Available since version 1.7.

I_VVSV_STRESSES_TORSION : = 0x0200
Available since version 1.7.

I_VVSV_DEFORMATION_EXACT : = 0x0001
Available since version 1.7.

I_VVSV_DEFORMATION_STD : = 0x0002
Available since version 1.7.

I_VVSV_REACTIONS_F : = 0x01
Available since version 1.7.

I_VVSV_REACTIONS_M : = 0x02
Available since version 1.7.

I_VVSV_REACTIONS_RESID_F : = 0x04
Available since version 1.7.

I_VVSV_REACTIONS_RESID_M : = 0x08
Available since version 1.7.

I_VVSV_REACTIONS_VAL : = 0x10
Available since version 1.7.

I_VVSV_REACTIONS_ROT : = 0x20
Available since version 1.7.

I_VVSV_REACTIONS_DIV : = 0x40
Available since version 1.7.

I_VVSV_LOADS_SYMBOL : = 0x01
Available since version 1.7.

I_VVSV_LOADS_VALUE : = 0x02
Available since version 1.7.

I_VVSV_STRESSES_GLOBAL_USER_MIN : = 0x001
Available since version 1.7.

I_VVSV_STRESSES_GLOBAL_USER_MAX : = 0x002
Available since version 1.7.

I_VVSV_STRESSES_GLOBAL_NORMAL_MIN : = 0x004
Available since version 1.7.

I_VVSV_STRESSES_GLOBAL_NORMAL_MAX : = 0x008
Available since version 1.7.

I_VVSV_STRESSES_GLOBAL_TAU_MIN : = 0x010
Available since version 1.7.

I_VVSV_STRESSES_GLOBAL_TAU_MAX : = 0x020
Available since version 1.7.

I_VVSV_STRESSES_GLOBAL_MISES_MIN : = 0x040
Available since version 1.7.

I_VVSV_STRESSES_GLOBAL_MISES_MAX : = 0x080
Available since version 1.7.

I_VVSV_BAR_MAPS_FX : = 165
Available since version 1.7.

I_VVSV_BAR_MAPS_FY : = 166
Available since version 1.7.

I_VVSV_BAR_MAPS_FZ : = 167
Available since version 1.7.

I_VVSV_BAR_MAPS_MX : = 168
Available since version 1.7.

I_VVSV_BAR_MAPS_MY : = 169
Available since version 1.7.

I_VVSV_BAR_MAPS_MZ : = 170
Available since version 1.7.

I_VVSV_BAR_MAPS_STRESS_S_MAX : = 171
Available since version 1.7.

I_VVSV_BAR_MAPS_STRESS_S_MIN : = 172
Available since version 1.7.

I_VVSV_BAR_MAPS_STRESS_FX_AX : = 173
Available since version 1.7.

I_VVSV_BAR_MAPS_STRESS_S_MAX_MY : = 174
Available since version 1.7.

I_VVSV_BAR_MAPS_STRESS_S_MAX_MZ : = 175
Available since version 1.7.

I_VVSV_BAR_MAPS_STRESS_S_MIN_MY : = 176
Available since version 1.7.

I_VVSV_BAR_MAPS_STRESS_S_MIN_MZ : = 177
Available since version 1.7.

I_VVSV_BAR_MAPS_SHEAR_STRESS_TY : = 179
Available since version 1.7.

I_VVSV_BAR_MAPS_SHEAR_STRESS_TZ : = 180
Available since version 1.7.

I_VVSV_BAR_MAPS_SHEAR_STRESS_T : = 181
Available since version 1.7.

I_VVSV_BAR_MAPS_DESIGN_RATIO : = 796
Available since version 1.7.

I_VVSV_BAR_MAPS_DESIGN_MEMB_LENGTH : = 271
Available since version 1.7.

I_VVSV_BAR_MAPS_DESIGN_SLEND_LAY : = 831
Available since version 1.7.

I_VVSV_BAR_MAPS_DESIGN_SLEND_LAZ : = 832
Available since version 1.7.

I_VVSV_DESCRIPTIONS_NONE : = 0
Available since version 2.0.

I_VVSV_DESCRIPTIONS_HORIZONTAL : = 0x01
Available since version 2.0.

I_VVSV_DESCRIPTIONS_VERTICAL : = 0x02
Available since version 2.0.

I_VVSV_REINFORCEMENT_A1 : = 0x0001
Available since version 2.0.

I_VVSV_REINFORCEMENT_A1R : = 0x0010
Available since version 2.0.

I_VVSV_REINFORCEMENT_A2 : = 0x0002
Available since version 2.0.

I_VVSV_REINFORCEMENT_A2R : = 0x0020
Available since version 2.0.

I_VVSV_REINFORCEMENT_PRC : = 0x0004
Available since version 2.0.

I_VVSV_REINFORCEMENT_PRCR : = 0x0008
Available since version 2.0.

I_VVSV_REINFORCEMENT_DISTRIB : = 0x0040
Available since version 2.0.

I_VVSV_REINFORCEMENT_DISTRIBR : = 0x0080
Available since version 2.0.

I_VVSV_REINFORCEMENT_A1BARS : = 0x0100
Available since version 2.0.

I_VVSV_REINFORCEMENT_A2BARS : = 0x0200
Available since version 2.0.

I_VVSV_MAPS_DETAILED_STRESSES_XX : = 483
Available since version 2.0.

I_VVSV_MAPS_DETAILED_STRESSES_YY : = 484
Available since version 2.0.

I_VVSV_MAPS_DETAILED_STRESSES_XY : = 485
Available since version 2.0.

I_VVSV_MAPS_DETAILED_STRESSES_Z : = 486
Available since version 2.0.

I_VVSV_MAPS_DETAILED_MEMB_FORCES_XX : = 492
Available since version 2.0.

I_VVSV_MAPS_DETAILED_MEMB_FORCES_YY : = 493
Available since version 2.0.

I_VVSV_MAPS_DETAILED_MEMB_FORCES_XY : = 494
Available since version 2.0.

I_VVSV_MAPS_DETAILED_MOMENTS_XX : = 501
Available since version 2.0.

I_VVSV_MAPS_DETAILED_MOMENTS_YY : = 502
Available since version 2.0.

I_VVSV_MAPS_DETAILED_MOMENTS_XY : = 503
Available since version 2.0.

I_VVSV_MAPS_DETAILED_SHEAR_STRESSES_XX : = 510
Available since version 2.0.

I_VVSV_MAPS_DETAILED_SHEAR_STRESSES_YY : = 511
Available since version 2.0.

I_VVSV_MAPS_DETAILED_SHEAR_FORCES_XX : = 519
Available since version 2.0.

I_VVSV_MAPS_DETAILED_SHEAR_FORCES_YY : = 520
Available since version 2.0.

I_VVSV_MAPS_DETAILED_DISPLACEMENTS_XX : = 537
Available since version 2.0.

I_VVSV_MAPS_DETAILED_DISPLACEMENTS_YY : = 538
Available since version 2.0.

I_VVSV_MAPS_DETAILED_DISPLACEMENTS_Z : = 531
Available since version 2.0.

I_VVSV_MAPS_DETAILED_ROTATION_XX : = 546
Available since version 2.0.

I_VVSV_MAPS_DETAILED_ROTATION_YY : = 547
Available since version 2.0.

I_VVSV_MAPS_DETAILED_ROTATION_Z : = 549
Available since version 2.0.

I_VVSV_MAPS_DETAILED_SOIL_REACTIONS : = 558
Available since version 2.0.

I_VVSV_MAPS_PRINCIPAL_STRESSES_1 : = 487
Available since version 2.0.

I_VVSV_MAPS_PRINCIPAL_STRESSES_2 : = 488
Available since version 2.0.

I_VVSV_MAPS_PRINCIPAL_STRESSES_1_2 : = 490
Available since version 2.0.

I_VVSV_MAPS_PRINCIPAL_STRESSES_ANGLE : = 489
Available since version 2.0.

I_VVSV_MAPS_PRINCIPAL_MEMB_FORCES_1 : = 496
Available since version 2.0.

I_VVSV_MAPS_PRINCIPAL_MEMB_FORCES_2 : = 497
Available since version 2.0.

I_VVSV_MAPS_PRINCIPAL_MEMB_FORCES_1_2 : = 499
Available since version 2.0.

I_VVSV_MAPS_PRINCIPAL_MEMB_FORCES_ANGLE : = 498
Available since version 2.0.

I_VVSV_MAPS_PRINCIPAL_MOMENTS_1 : = 505
Available since version 2.0.

I_VVSV_MAPS_PRINCIPAL_MOMENTS_2 : = 506
Available since version 2.0.

I_VVSV_MAPS_PRINCIPAL_MOMENTS_ANGLE : = 507
Available since version 2.0.

I_VVSV_MAPS_PRINCIPAL_MOMENTS_1_2 : = 508
Available since version 2.0.

I_VVSV_MAPS_PRINCIPAL_SHEAR_STRESSES : = 517
Available since version 2.0.

I_VVSV_MAPS_PRINCIPAL_SHEAR_FORCES : = 526
Available since version 2.0.

I_VVSV_MAPS_PRINCIPAL_GLOBAL_DISP_X : = 1302
Available since version 2.0.

I_VVSV_MAPS_PRINCIPAL_GLOBAL_DISP_Y : = 1303
Available since version 2.0.

I_VVSV_MAPS_PRINCIPAL_GLOBAL_DISP_Z : = 1304
Available since version 2.0.

I_VVSV_MAPS_PRINCIPAL_TOTAL_DISP : = 1305
Available since version 2.0.

I_VVSV_MAPS_COMPLEX_STRESSES : = 491
Available since version 2.0.

I_VVSV_MAPS_COMPLEX_MEMB_FORCES : = 500
Available since version 2.0.

I_VVSV_MAPS_COMPLEX_MOMENTS : = 509
Available since version 2.0.

I_VVSV_MAPS_COMPLEX_REINFORCEMENT_TOP_MXX : = 889
Available since version 2.0.

I_VVSV_MAPS_COMPLEX_REINFORCEMENT_TOP_MYY : = 891
Available since version 2.0.

I_VVSV_MAPS_COMPLEX_REINFORCEMENT_BOTTOM_MXX : = 890
Available since version 2.0.

I_VVSV_MAPS_COMPLEX_REINFORCEMENT_BOTTOM_MYY : = 892
Available since version 2.0.

I_VVSV_MAPS_CROSS_S : = 1404
Available since version 2.0.

I_VVSV_MAPS_CROSS_N : = 1405
Available since version 2.0.

I_VVSV_MAPS_CROSS_M : = 1406
Available since version 2.0.

I_VVSV_MAPS_LAYER_UPPER : = 0
Available since version 2.0.

I_VVSV_MAPS_LAYER_MIDDLE : = 1
Available since version 2.0.

I_VVSV_MAPS_LAYER_LOWER : = 2
Available since version 2.0.

I_VVSV_MAPS_LAYER_MAXIMUM : = 3
Available since version 2.0.

I_VVSV_MAPS_LAYER_MINIMUM : = 4
Available since version 2.0.

I_VVSV_MAPS_LAYER_ABS_MINIMUM : = 5
Available since version 2.0.

I_VVSV_MAPS_LAYER_ARBITRARY : = 6
Available since version 2.0.

I_VVSV_MAPS_ISOLINES : = 0x0001
Available since version 2.0.

I_VVSV_MAPS_DESCRIPTION : = 0x0002
Available since version 2.0.

I_VVSV_MAPS_SMOOTH_NON : = 0x0004
Available since version 2.0.

I_VVSV_MAPS_SMOOTH_GLO : = 0x0008
Available since version 2.0.

I_VVSV_MAPS_SMOOTH_LOC : = 0x0010
Available since version 2.0.

I_VVSV_MAPS_SMOOTH_SEL : = 0x0020
Available since version 2.0.

I_VVSV_MAPS_SMOOTH_CAR : = 0x0030
Available since version 2.0.

I_VVSV_FORCES_BAR_REACT_KY : = 0x0080

Available since version 3.

I_VVSV_FORCES_BAR_REACT_KZ : = 0x0800

Available since version 3.

I_VVSV_FILL_FE : = 0x80
finite element mesh
Available since version 4.1.

IRobotViewScaleType

Available types of a scale defined for a graphical viewer.

Properties:

I_VST_FX : = 0
a scale for FX force

I_VST_FY : = 1
a scale for FY force

I_VST_FZ : = 2
a scale for FZ force

I_VST_MX : = 3
a scale for MX

I_VST_MY : = 4
a scale for MY

I_VST_MZ : = 5
a scale for MZ

I_VST_DEFORM : = 6
a scale for deformation

I_VST_SIG : = 7
a scale for axial stresses

I_VST_TAU : = 8
a scale for shear stresses

IRobotViewDetailedAnalysis

Derived from IRobotView
View presenting results of the detailed analysis.
Available since version 2.0.

Properties:

CurrentTableTab : IRobotViewDetailedAnalysisTableTab
identifier of the active tab in the tabular part of the view

ParamsDetailed : IRobotViewDetailedAnalysisParams
parameters of displaying detailed analysis results
Available since version 2.0.

Window : IRobotWindow

Available since version 3.

UserControl : bool
flag that enables giving/taking back graphical view navigation to/from the interactive program user

Available since version 3.

Operations:

MakeScreenCapture (_sc_params : IRobotViewScreenCaptureParams)

IRobotViewDetailedAnalysisTableTab

Identifiers of the tabs in the table being a part of the view presenting detailed analysis results.
Available since version 2.0.

Properties:

I_VDATT_VALUES : = 0
I_VDATT_LOCAL_EXTREMES : = 1
tab presenting local extreme values

I_VDATT_GLOBAL_EXTREMES : = 2

tab presenting global extreme values

IRobotViewDisplayParams

Parameters of display for graphical viewers

Properties:

HiddenLines : IRobotViewHiddenLinesDisplayType
type of displaying hidden lines
Available since version 2.0.

SymbolSize : short
symbol size (from 1 to 10)
Available since version 2.0.

Operations:

Set (v_attrib : IRobotViewDisplayAttributes, on_off : bool)
Function switches on/off a selected visibility attribute.
Available since version 2.0.

IsOn (v_attrib : IRobotViewDisplayAttributes) : bool
Function returns a value different from zero (True) if a selected attribute is switched on.
Available since version 2.0.

IRobotViewDisplayAttributes

Set of attributes whose visibility may be switched on/off for graphical views.

Properties:

I_VDA_STRUCTURE_STRUCTURE : = 1
Available since version 2.0.

I_VDA_STRUCTURE_NODE_NUMBERS : = 2
Available since version 2.0.

I_VDA_STRUCTURE_BAR_NUMBERS : = 3
Available since version 2.0.

I_VDA_STRUCTURE_SUPPORT_SYMBOLS : = 4
Available since version 2.0.

I_VDA_STRUCTURE_SUPPORT_CODES : = 5
Available since version 2.0.

I_VDA_STRUCTURE_ONLY_FOR_SELECTED_OBJECTS : = 6

Available since version 2.0.

I_VDA_STRUCTURE_EXPLODE : = 7
Available since version 2.0.

I_VDA_STRUCTURE_GROUP_COLORS : = 8
Available since version 2.0.

I_VDA_STRUCTURE_ATTRIBUTE_LABELS : = 9
Available since version 2.0.

I_VDA_STRUCTURE_LOCAL_SYSTEM : = 10
Available since version 2.0.

I_VDA_SECTIONS_NAME : = 11
Available since version 2.0.

I_VDA_SECTIONS_COLORS : = 12
Available since version 2.0.

I_VDA_SECTIONS_SYMBOLS : = 13
Available since version 2.0.

I_VDA_SECTIONS_SHAPE : = 14
Available since version 2.0.

I_VDA_SECTIONS_SURFACE : = 15
Available since version 2.0.

I_VDA_SECTIONS_MATERIAL : = 16
Available since version 2.0.

I_VDA_LOADS_SYMBOLS : = 17
Available since version 2.0.

I_VDA_LOADS_VALUES : = 18
Available since version 2.0.

I_VDA_LOADS_MOVING_ROUTE : = 19
Available since version 2.0.

I_VDA_LOADS_MOVING_VEHICLE : = 20
Available since version 2.0.

I_VDA_LOADS_MOVING_ELEMENTS : = 21
Available since version 2.0.

I_VDA_LOADS_MOVING_LOADS : = 22
Available since version 2.0.

I_VDA_ADVANCED_RELEASE_SYMBOLS : = 23

Available since version 2.0.

I_VDA_ADVANCED_RELEASE_CODES : = 24
Available since version 2.0.

I_VDA_ADVANCED_OFFSETS : = 25
Available since version 2.0.

I_VDA_ADVANCED_COMPATIBLE_NODES : = 26
Available since version 2.0.

I_VDA_ADVANCED_RIGID_LINKS : = 27
Available since version 2.0.

I_VDA_ADVANCED_RIGID_LINK_SURFACE : = 28
Available since version 2.0.

I_VDA_ADVANCED_ELASTIC_FOUNDATION : = 29
Available since version 2.0.

I_VDA_ADVANCED_CABLES : = 30
Available since version 2.0.

I_VDA_ADVANCED_TENSION_COMPRESSION : = 31
Available since version 2.0.

I_VDA_ADVANCED_STEEL_CONNECTIONS : = 32
Available since version 2.0.

I_VDA_FE_PANEL_CONTOURS : = 33
Available since version 2.0.

I_VDA_FE_PANEL_INTERIOR : = 34
Available since version 2.0.

I_VDA_FE_THICKNESS : = 35
Available since version 2.0.

I_VDA_FE_PANEL_NUMBERS : = 36
Available since version 2.0.

I_VDA_FE_PANEL_COMPLEX_DESC : = 37
Available since version 2.0.

I_VDA_FE_COLOR_LEGEND : = 38
Available since version 2.0.

I_VDA_FE_CONTOUR_COMPONENTS : = 39
Available since version 2.0.

I_VDA_FE_MESH_PREVIEW : = 40

Available since version 2.0.

I_VDA_FE_EDGE_NUMBERS : = 41
Available since version 2.0.

I_VDA_FE_FINITE_ELEMENTS : = 42
Available since version 2.0.

I_VDA_FE_FINITE_ELEMENT_NUMBERS : = 43
Available since version 2.0.

I_VDA_FE_EMITTERS : = 44
Available since version 2.0.

I_VDA_OTHER_STRUCTURAL_AXIS : = 45
Available since version 2.0.

I_VDA_OTHER_DIMENSION_LINES : = 46
Available since version 2.0.

I_VDA_OTHER_GRID : = 47
Available since version 2.0.

I_VDA_OTHER_RULER : = 48
Available since version 2.0.

I_VDA_OTHER_HIDE_NODES : = 49
Available since version 2.0.

I_VDA_OTHER_INACTIVE : = 50
Available since version 2.0.

I_VDA_OTHER_HIDE_INACTIVE : = 51
Available since version 2.0.

I_VDA_OTHER_CALC_ELEM_NUMBERS : = 52
Available since version 2.0.

I_VDA_OTHER_CALC_POINT_NUMBERS : = 53
Available since version 2.0.

I_VDA_OTHER_OBJECTS_OUT_OF_PLANE : = 54
Available since version 2.0.

I_VDA_SECTIONS_MEMBER_TYPE_NAME : = 55
bar type - name
Available since version 3.5.

I_CDA_SECTIONS_MEMBER_TYPE_LEGEND : = 56
bar type - color legend
Available since version 3.5.

I_VDA_ADVANCED_GEOIMPERFECTIONS : = 57
geometrical imperfections
Available since version 3.5.

I_VDA_ADVANCED_GEOIMPERFECTIONS_NAME : = 58
geometrical imperfections - name
Available since version 3.5.

I_VDA_ADVANCED_NONLINEAR_HINGES : = 59
non-linear hinges
Available since version 3.5.

IRobotViewHiddenLinesDisplayType

Possible types of displaying hidden lines.

Properties:

I_VHLDT_HIDDEN_LINES : = 0x04
Available since version 2.0.

I_VHLDT_SHADING : = 0x10
Available since version 2.0.

I_VHLDT_COLOR_SIDES_EDGES : = 0x40
Available since version 2.0.

I_VHLDT_SHADING_EDGES : = 0x80
Available since version 2.0.

I_VHLDT_QUICK_SHADING_OF_FACES : = 0x100
Available since version 2.0.

I_VHLDT_NONE : = 0
Available since version 2.0.

IRobotViewDiagramParams

Interface controlling the manner of displaying diagrams on bars.

Properties:

ReactionsInLocalSystem : bool
Available since version 2.0.

Descriptions : IRobotViewDiagramDescriptionType
the manner of locating the diagram description
Available since version 2.0.

PositiveNegative : IRobotViewDiagramSignDifferType
manner of differentiating the values of different signs on the diagrams
Available since version 2.0.

Filling : IRobotViewDiagramFillingType
Type of diagram filling
Available since version 2.0.

Operations:

Set (res_type : IRobotViewDiagramResultType, on_off : bool)
Function switches on/off diagram display for a specified result type.
Available since version 2.0.

IsOn (res_type : IRobotViewDiagramResultType) : bool
Function returns value different from zero (True) if a specified result type is switched on (i.e. it is displayed in the view).
Available since version 2.0.

SetScale (res_type : IRobotViewDiagramResultType, scale_for_1_cm : double)
Function sets a diagram scale for 1 cm for a specified result type. If the same scale is valid for several associated results it is set for all of them. If scale definition for a specified result type makes no sense, then scale value is ignored.
Available since version 2.0.

GetScale (res_type : IRobotViewDiagramResultType) : double
Function returns scale value for a specified result diagram for 1 cm.
Available since version 2.0.

SetColor (res_type : IRobotViewDiagramResultType, _color : long)
Function sets color which is to be used while drawing a diagram of a specified result type.
Available since version 2.0.

GetColor (res_type : IRobotViewDiagramResultType) : long
Function returns color which is used while drawing a diagram of a specified result type.
Available since version 2.0.

Values : IRobotViewDiagramValueType

Available since version 8.1.

IRobotViewDiagramResultType

Set of result types presented by means of diagrams on bars.

Properties:

I_VDRT_NTM_FX : = 1
Available since version 2.0.

I_VDRT_NTM_FY : = 2
Available since version 2.0.

I_VDRT_NTM_FZ : = 3
Available since version 2.0.

I_VDRT_NTM_MX : = 4
Available since version 2.0.

I_VDRT_NTM_MY : = 5
Available since version 2.0.

I_VDRT_NTM_MZ : = 6
Available since version 2.0.

I_VDRT_DEFORMATION_DEFORMATION : = 7
Available since version 2.0.

I_VDRT_DEFORMATION_EXACT : = 8
Available since version 2.0.

I_VDRT_STRESS_S_MAX : = 9
Available since version 2.0.

I_VDRT_STRESS_S_MIN : = 10
Available since version 2.0.

I_VDRT_STRESS_BENDING_S_MAX_MY : = 11
Available since version 2.0.

I_VDRT_STRESS_BENDING_S_MAX_MZ : = 12
Available since version 2.0.

I_VDRT_STRESS_BENDING_S_MIN_MY : = 13
Available since version 2.0.

I_VDRT_STRESS_BENDING_S_MIN_MZ : = 14
Available since version 2.0.

I_VDRT_STRESS_AXIAL_FX_AX : = 15
Available since version 2.0.

I_VDRT_STRESS_SHEAR_TY : = 16
Available since version 2.0.

I_VDRT_STRESS_SHEAR_TZ : = 17
Available since version 2.0.

I_VDRT_STRESS_TORSION : = 18
Available since version 2.0.

I_VDRT_REACTION_FORCES : = 19
Available since version 2.0.

I_VDRT_REACTION_MOMENTS : = 20
Available since version 2.0.

I_VDRT_REACTION_RESIDUAL_FORCES : = 21
Available since version 2.0.

I_VDRT_REACTION_RESIDUAL_MOMENTS : = 22
Available since version 2.0.

I_VDRT_REINFORCE_TOP : = 23
Available since version 2.0.

I_VDRT_REINFORCE_TOP_REAL : = 24
Available since version 2.0.

I_VDRT_REINFORCE_BOTTOM : = 25
Available since version 2.0.

I_VDRT_REINFORCE_BOTTOM_REAL : = 26
Available since version 2.0.

I_VDRT_REINFORCE_RATIO : = 27
Available since version 2.0.

I_VDRT_REINFORCE_RATIO_REAL : = 28
Available since version 2.0.

I_VDRT_REINFORCE_STIRRUP_SPACING : = 29
Available since version 2.0.

I_VDRT_REINFORCE_STIRRUP_SPACING_REAL : = 30
Available since version 2.0.

I_VDRT_REINFORCE_NUMBER_OF_UPPER_BARS : = 31
Available since version 2.0.

I_VDRT_REINFORCE_NUMBER_OF_LOWER_BARS : = 32
Available since version 2.0.

I_VDRT_NTM_KY : = 33
coefficient of elastic ground reaction in Y direction
Available since version 3.

I_VDRT_NTM_KZ : = 34
coefficient of elastic ground reaction in Z direction

Available since version 3.

IRobotViewDiagramDescriptionType

Available methods of locating descriptions for diagrams.

Properties:

I_VDDT_NONE : = 0
Available since version 2.0.

I_VDDT_HORIZONTAL : = 1
Available since version 2.0.

I_VDDT_VERTICAL : = 2
Available since version 2.0.

I_VDDT_LABELS : = 1

Available since version 8.1.

I_VDDT_TEXT : = 2

Available since version 8.1.

IRobotViewDiagramFillingType

Available methods of diagram filling.

Properties:

I_VDFT_FENCE : = 0
Available since version 2.0.

I_VDFT_FILLED : = 1
Available since version 2.0.

IRobotViewDiagramSignDifferType

Available methods of distinguishing positive and negative values on diagrams

Properties:

I_VDSDT_UNDIFFERENTIATED : = 0
Available since version 2.0.

I_VDSDT_DIFFERENTIATED : = 1
Available since version 2.0.

IRobotViewDiagrams

Derived from IRobotView3
A viewer dedicated to result presentation.

IRobotViewBarMaps

Derived from IRobotView3
View presenting maps on bars with a scale.

IRobotViewFeMaps

Derived from IRobotView3
View presenting maps on finite elements with a scale.

IRobotViewBarMapResultType

Set of result types presented by means of maps on bars.

Properties:

I_VBMRT_NTM_FX : = 165
Available since version 2.0.

I_VBMRT_NTM_FY : = 166
Available since version 2.0.

I_VBMRT_NTM_FZ : = 167
Available since version 2.0.

I_VBMRT_NTM_MX : = 168
Available since version 2.0.

I_VBMRT_NTM_MY : = 169
Available since version 2.0.

I_VBMRT_NTM_MZ : = 170
Available since version 2.0.

I_VBMRT_STRESS_S_MAX : = 171
Available since version 2.0.

I_VBMRT_STRESS_S_MIN : = 172
Available since version 2.0.

I_VBMRT_STRESS_S_MAX_MY : = 174
Available since version 2.0.

I_VBMRT_STRESS_S_MIN_MY : = 176
Available since version 2.0.

I_VBMRT_STRESS_S_MAX_MZ : = 175
Available since version 2.0.

I_VBMRT_STRESS_S_MIN_MZ : = 177
Available since version 2.0.

I_VBMRT_STRESS_FX_AX : = 173
Available since version 2.0.

I_VBMRT_STRESS_SHEAR_TY : = 179
Available since version 2.0.

I_VBMRT_STRESS_SHEAR_TZ : = 180
Available since version 2.0.

I_VBMRT_STRESS_TORSION_T : = 182
Available since version 2.0.

I_VBMRT_DESIGN_RATIO : = 796
Available since version 2.0.

I_VBMRT_DESIGN_MEMBER_LENGTH : = 271
Available since version 2.0.

I_VBMRT_DESIGN_SLEND_LAY : = 831
Available since version 2.0.

I_VBMRT_DESIGN_SLEND_LAZ : = 832
Available since version 2.0.

I_VBMRT_NOTHING : = 0
Available since version 2.0.

IRobotViewBarMapParams

Parameters describing the manner of displaying maps on bars.

Properties:

CurrentResult : IRobotViewBarMapResultType
result type for which a map on bars is currently displayed

StructureDeformation : bool
flag switching on/off structure deformation drawing
Available since version 2.0.

Descriptions : IRobotViewDiagramDescriptionType
manner of displaying map description
Available since version 2.0.

MapThicknessCoeff : double
thickness coefficient of a map drawing line in relation to the thickness of a bar drawing line
Available since version 2.0.

IRobotViewDetailedAnalysisResultType

Result types presented by detailed analysis view.

Properties:

I_VDART_NTM_FX : = 1
Available since version 2.0.

I_VDART_NTM_FY : = 2
Available since version 2.0.

I_VDART_NTM_FZ : = 3
Available since version 2.0.

I_VDART_NTM_MX : = 4
Available since version 2.0.

I_VDART_NTM_MY : = 5
Available since version 2.0.

I_VDART_NTM_MZ : = 6
Available since version 2.0.

I_VDART_NTM_UX : = 7
Available since version 2.0.

I_VDART_NTM_UY : = 8
Available since version 2.0.

I_VDART_NTM_UZ : = 9
Available since version 2.0.

I_VDART_STRESS_S_MAX : = 10
Available since version 2.0.

I_VDART_STRESS_S_MIN : = 11
Available since version 2.0.

I_VDART_STRESS_BENDING_S_MAX_MY : = 12
Available since version 2.0.

I_VDART_STRESS_BENDING_S_MIN_MY : = 13
Available since version 2.0.

I_VDART_STRESS_BENDING_S_MAX_MZ : = 14
Available since version 2.0.

I_VDART_STRESS_BENDING_S_MIN_MZ : = 15
Available since version 2.0.

I_VDART_STRESS_AXIAL_FX_AX : = 16
Available since version 2.0.

I_VDART_STRESS_SHEAR_TY : = 17
Available since version 2.0.

I_VDART_STRESS_SHEAR_TZ : = 18
Available since version 2.0.

I_VDART_STRESS_TORSION_T : = 19
Available since version 2.0.

I_VDART_REINFORCE_TOP : = 20
Available since version 2.0.

I_VDART_REINFORCE_TOP_REAL : = 21
Available since version 2.0.

I_VDART_REINFORCE_BOTTOM : = 22
Available since version 2.0.

I_VDART_REINFORCE_BOTTOM_REAL : = 23
Available since version 2.0.

I_VDART_REINFORCE_RATIO : = 24
Available since version 2.0.

I_VDART_REINFORCE_RATIO_REAL : = 25
Available since version 2.0.

I_VDART_REINFORCE_STIRRUP_SPACING : = 26
Available since version 2.0.

I_VDART_REINFORCE_STIRRUP_SPACING_REAL : = 27
Available since version 2.0.

I_VDART_REINFORCE_NUMBER_OF_UPPER_BARS : = 28
Available since version 2.0.

I_VDART_REINFORCE_NUMBER_OF_LOWER_BARS : = 29
Available since version 2.0.

IRobotViewDetailedAnalysisParams

Parameters of detailed analysis view.

Properties:

Descriptions : IRobotViewDiagramDescriptionType
manner of displaying diagram description
Available since version 2.0.

PositiveNegative : IRobotViewDiagramSignDifferType
manner of presenting values of different signs on the diagram
Available since version 2.0.

Filling : IRobotViewDiagramFillingType
type of diagram filling
Available since version 2.0.

ReinforceShowTheoreticAndRealVals : bool
Available since version 2.0.

Operations:

Set (res_type : IRobotViewDetailedAnalysisResultType, on_off : bool)
Function switches on/off display of a defined result type.
Available since version 2.0.

IsOn (res_type : IRobotViewDetailedAnalysisResultType) : bool
Function returns value different from zero (True) if a selected result type is displayed by the detailed analysis view.
Available since version 2.0.

SetColor (res_type : IRobotViewDetailedAnalysisResultType, _color : long)
Function sets color which is to be used while drawing a diagram for a specified result type.
Available since version 2.0.

GetColor (res_type : IRobotViewDetailedAnalysisResultType) : long
Function returns color assigned to a specified result type.
Available since version 2.0.

IRobotViewFeMapResultType

Result types presented by means of maps on finite elements.

Properties:

I_VFMRT_DETAILED_STRESS_XX : = 483

Available since version 2.0.

I_VFMRT_DETAILED_STRESS_YY : = 484
Available since version 2.0.

I_VFMRT_DETAILED_STRESS_XY : = 485
Available since version 2.0.

I_VFMRT_DETAILED_STRESS_Z : = 486
Available since version 2.0.

I_VFMRT_DETAILED_MEMBRANE_FORCE_XX : = 492
Available since version 2.0.

I_VFMRT_DETAILED_MEMBRANE_FORCE_YY : = 493
Available since version 2.0.

I_VFMRT_DETAILED_MEMBRANE_FORCE_XY : = 494
Available since version 2.0.

I_VFMRT_DETAILED_MOMENT_XX : = 501
Available since version 2.0.

I_VFMRT_DETAILED_MOMENT_YY : = 502
Available since version 2.0.

I_VFMRT_DETAILED_MOMENT_XY : = 503
Available since version 2.0.

I_VFMRT_DETAILED_SHEAR_STRESS_XX : = 510
Available since version 2.0.

I_VFMRT_DETAILED_SHEAR_STRESS_YY : = 511
Available since version 2.0.

I_VFMRT_DETAILED_SHEAR_FORCE_XX : = 519
Available since version 2.0.

I_VFMRT_DETAILED_SHEAR_FORCE_YY : = 520
Available since version 2.0.

I_VFMRT_DETAILED_DISPLACEMENT_XX : = 537
Available since version 2.0.

I_VFMRT_DETAILED_DISPLACEMENT_YY : = 538
Available since version 2.0.

I_VFMRT_DETAILED_ROTATION_XX : = 546
Available since version 2.0.

I_VFMRT_DETAILED_ROTATION_YY : = 547

Available since version 2.0.

I_VFMRT_DETAILED_ROTATION_Z : = 549
Available since version 2.0.

I_VFMRT_DETAILED_SOIL_REACTION_Z : = 558
Available since version 2.0.

I_VFMRT_PRINCIPAL_STRESS_1 : = 487
Available since version 2.0.

I_VFMRT_PRINCIPAL_STRESS_2 : = 488
Available since version 2.0.

I_VFMRT_PRINCIPAL_STRESS_1_2 : = 490
Available since version 2.0.

I_VFMRT_PRINCIPAL_STRESS_ANGLE : = 489
Available since version 2.0.

I_VFMRT_PRINCIPAL_MEMBRANE_FORCE_1 : = 496
Available since version 2.0.

I_VFMRT_PRINCIPAL_MEMBRANE_FORCE_2 : = 497
Available since version 2.0.

I_VFMRT_PRINCIPAL_MEMBRANE_FORCE_1_2 : = 499
Available since version 2.0.

I_VFMRT_PRINCIPAL_MEMBRANE_FORCE_ANGLE : = 498
Available since version 2.0.

I_VFMRT_PRINCIPAL_MOMENT_1 : = 505
Available since version 2.0.

I_VFMRT_PRINCIPAL_MOMENT_2 : = 506
Available since version 2.0.

I_VFMRT_PRINCIPAL_MOMENT_1_2 : = 508
Available since version 2.0.

I_VFMRT_PRINCIPAL_MOMENT_ANGLE : = 507
Available since version 2.0.

I_VFMRT_PRINCIPAL_SHEAR_STRESS_1_2 : = 517
Available since version 2.0.

I_VFMRT_PRINCIPAL_SHEAR_FORCE_1_2 : = 526
Available since version 2.0.

I_VFMRT_GLOBAL_DISPLACEMENT_X : = 1302

Available since version 2.0.

I_VFMRT_GLOBAL_DISPLACEMENT_Y : = 1303
Available since version 2.0.

I_VFMRT_GLOBAL_DISPLACEMENT_Z : = 1304
Available since version 2.0.

I_VFMRT_TOTAL_DISPLACEMENTS : = 313058
Available since version 2.0.

I_VFMRT_COMPLEX_STRESSES : = 491
Available since version 2.0.

I_VFMRT_COMPLEX_MEMBRANE_FORCES : = 500
Available since version 2.0.

I_VFMRT_COMPLEX_MOMENTS : = 509
Available since version 2.0.

I_VFMRT_COMPLEX_REINFORCE_TOP_MXX : = 889
Available since version 2.0.

I_VFMRT_COMPLEX_REINFORCE_TOP_MYY : = 891
Available since version 2.0.

I_VFMRT_COMPLEX_REINFORCE_BOTTOM_MXX : = 890
Available since version 2.0.

I_VFMRT_COMPLEX_REINFORCE_BOTTOM_MYY : = 892
Available since version 2.0.

IRobotViewFeMapParams

Parameters of presenting maps on finite elements.

Properties:

CurrentResult : IRobotViewFeMapResultType
currently displayed result type
Available since version 2.0.

Direction : IRobotViewFeMapLocalSystemType
Available since version 2.0.

DirectionData : IRobotGeoPoint3D
additional data describing direction;
depending on the selected method of determining direction, it contains coordinates of the point or vector defining the direction (for the polar system defined in a node, the DirectionNode variable denoting a node number should be applied)

Available since version 2.0.

CrossPresentation : IRobotViewFeMapCrossPresentationType
Available since version 2.0.

DeformationActive : bool
Available since version 2.0.

DeformationConstScale : bool
Available since version 2.0.

MapScale : double
Available since version 2.0.

Layer : IRobotViewFeMapLayerType
Available since version 2.0.

LayerArbitraryVal : double
Available since version 2.0.

Smoothing : IRobotViewFeMapSmoothingType
Available since version 2.0.

Isolines : bool
Available since version 2.0.

WithDescription : bool
Available since version 2.0.

DirectionNode : long
number of the node determining direction in case when direction is defined in the polar system by indicating a structure node
Available since version 2.0.

Operations:

SetDirectionData (_dir_data : IRobotGeoPoint3D)

Available since version 3.5.

IRobotViewFeMapLocalSystemType

Properties:

I_VFMLST_AUTOMATIC : = 0
Available since version 2.0.

I_VFMLST_CARTESIAN_ALONG_X : = 1
Available since version 2.0.

I_VFMLST_CARTESIAN_ALONG_Y : = 2
Available since version 2.0.

I_VFMLST_CARTESIAN_ALONG_Z : = 3
Available since version 2.0.

I_VFMLST_CARTESIAN_ALONG_VECTOR : = 4
Available since version 2.0.

I_VFMLST_POLAR_IN_NODE : = 5
Available since version 2.0.

I_VFMLST_POLAR_IN_POINT : = 6
Available since version 2.0.

IRobotViewFeMapCrossPresentationType

Manner of presenting crosses in the map view for surface finite elements.

Properties:

I_VFMCPT_NONE : = 0
Available since version 2.0.

I_VFMCPT_STRESSES : = 1404
Available since version 2.0.

I_VFMCPT_NORMAL_FORCES : = 1405
Available since version 2.0.

I_VFMCPT_MOMENTS : = 1406
Available since version 2.0.

IRobotViewFeMapLayerType

Properties:

I_VFMLT_UPPER : = 0
Available since version 2.0.

I_VFMLT_MIDDLE : = 1
Available since version 2.0.

I_VFMLT_LOWER : = 2
Available since version 2.0.

I_VFMLT_MAXIMUM : = 3

Available since version 2.0.

I_VFMLT_MINIMUM : = 4
Available since version 2.0.

I_VFMLT_ABSOLUTE_MAXIMUM : = 5
Available since version 2.0.

I_VFMLT_ARBITRARY : = 6
Available since version 2.0.

IRobotViewFeMapSmoothingType

Properties:

I_VFMST_NO_SMOOTHING : = 0
Available since version 2.0.

I_VFMST_GLOBAL_SMOOTHING : = 1
Available since version 2.0.

I_VFMST_SMOOTHING_WITH_PANEL : = 2
Available since version 2.0.

I_VFMST_SMOOTHING_ACCORDING_TO_SELECTION : = 3
Available since version 2.0.

IRobotView2

Derived from IRobotView
Extended graphical view interface.
Available since version 3.

Properties:

Window : IRobotWindow

Available since version 3.

ParamsPanelCut : IRobotViewPanelCutParams
parameters of displaying panel cuts
Available since version 3.

UserControl : bool
flag that enables giving/taking back graphical view navigation to/from the interactive program user

Available since version 3.

IRobotViewDiagramPositionType

Available diagram positions with respect to the panel plane.
Available since version 3.

Properties:

I_VDPT_NORMAL :
diagram perpendicular to the panel plane
Available since version 3.

I_VDPT_IN_PLANE :
diagram in the panel plane
Available since version 3.

IRobotViewReinforcementResultType

Available reinforcement results presented in the garphical view.
Available since version 3.

Properties:

I_VRRT_AX_TOP : = 577
Top reinforcement area (X direction)
Available since version 3.

I_VRRT_AX_BOTTOM : = 579
Bottom reinforcement area (X direction)
Available since version 3.

I_VRRT_AY_TOP : = 578
Top reinforcement area (Y direction)
Available since version 3.

I_VRRT_AY_BOTTOM : = 580
Bottom reinforcement area (Y direction)
Available since version 3.

I_VRRT_E_AX_TOP : = 699
Top reinforcement spacing (X direction)
Available since version 3.

I_VRRT_E_AX_BOTTOM : = 701
Bottom reinforcement spacing (X direction)
Available since version 3.

I_VRRT_E_AY_TOP : = 700
Top reinforcement spacing (Y direction)

Available since version 3.

I_VRRT_E_AY_BOTTOM : = 702
Bottom reinforcement spacing (Y direction)
Available since version 3.

I_VRRT_A_MIN : = 1119
minimal reinforcement area for one layer and one reinforcement direction
Available since version 3.

I_VRRT_AX : = 1116
Cracking (X axis direction)
Available since version 3.

I_VRRT_AY : = 1117
Cracking (Y axis direction)
Available since version 3.

I_VRRT_F : = 1118
deflection value
Available since version 3.

IRobotViewGlobalAnalysisParams

Interface controlling display of global analysis results.
Available since version 3.

Properties:

Results : IRobotViewGlobalAnalysisResultsParams
method of result display
Available since version 3.

Operations:

IsOn (_type : IRobotViewGlobalAnalysisParamsType) : bool
Function checks status of the indicated parameter.
Available since version 3.

Set (_type : IRobotViewGlobalAnalysisParamsType, _status : bool)
Function sets status for the indicated parameter.
Available since version 3.

IRobotViewGlobalAnalysisResultsParams

Interface controlling result display.
Available since version 3.

Properties:

Type : IRobotViewGlobalAnalysisResultsType
method of result presentation
Available since version 3.

NPointsValue : long
number of points along the bar length
Available since version 3.

RelativeValue : double
Relative point coordinate
Available since version 3.

IRobotViewGlobalAnalysisResultsType

Types of result display.
Available since version 3.

Properties:

I_VGART_N_POINTS : = 1

Available since version 3.

I_VGART_RELATIVE : = 2

Available since version 3.

IRobotViewGlobalAnalysisParamsType

Parameter types
Available since version 3.

Properties:

I_VGAPT_FORCE_FX : = 1

Available since version 3.

I_VGAPT_FORCE_FY : = 2

Available since version 3.

I_VGAPT_FORCE_FZ : = 3

Available since version 3.

I_VGAPT_FORCE_MX : = 4

Available since version 3.

I_VGAPT_FORCE_MY : = 5

Available since version 3.

I_VGAPT_FORCE_MZ : = 6

Available since version 3.

I_VGAPT_FORCE_KX : = 7

Available since version 3.

I_VGAPT_FORCE_KY : = 8

Available since version 3.

I_VGAPT_STRESS_NORMAL : = 9

Available since version 3.

I_VGAPT_STRESS_EXTREME : = 10

Available since version 3.

I_VGAPT_STRESS_BENDING_SMY : = 11

Available since version 3.

I_VGAPT_STRESS_BENDING_SMZ : = 12

Available since version 3.

I_VGAPT_STRESS_AXIAL_SXFX : = 13

Available since version 3.

I_VGAPT_STRESS_SHEAR : = 14

Available since version 3.

I_VGAPT_STRESS_TORSION : = 15

Available since version 3.

I_VGAPT_DESIGN_RATIO : = 16

Available since version 3.

I_VGAPT_DESIGN_MEMBER_LENGTH : = 17

Available since version 3.

I_VGAPT_DESIGN_SLENDERNESS_LAY : = 18

Available since version 3.

I_VGAPT_DESIGN_SLENDERNESS_LAZ : = 19

Available since version 3.

IRobotViewGlobalAnalysis

Derived from IRobotView2
View presenting global analysis results.
Available since version 3.

Properties:

ParamsGlobalAnalysis : IRobotViewGlobalAnalysisParams
display parameters for global analysis results
Available since version 3.

Operations:

MakeScreenCapture (_sc_params : IRobotViewScreenCaptureParams)

IRobotViewScreenCaptureParams

Parameters of a screen capture for the graphical view.

Properties:

Name : string
screen capture name

UpdateType : IRobotViewScreenCaptureUpdateType
Orientation : IRobotPageSetupOrientation
screen capture orientation (landscape or portait)

ScaleAutomatic : bool
True value denotes selection of the automatic scale, while False value indicates selection of the user scale defined by ScaleValue value

ScaleValue : double
user scale value - distance expressed in meters (m) which corresponds to one centimeter (1 cm) in the graphical view

IRobotViewScreenCaptureUpdateType

Available methods of updating the contents of a screen capture for the graphical view.

Properties:

I_SCUT_UPDATED_UPON_PRINTING : = 1
the view contents will be updated automatically while printing

I_SCUT_CURRENT_VIEW : = 2
the view will be printed in the form it had while making a screen capture - the view contents will not be updated

I_SCUT_UPDATED_WHOLE_STRUCTURE : = 3
the view contents will be updated while printing in such a way so that the view presents the whole current structure

IRobotView3

Derived from IRobotView2

Operations:

MakeScreenCapture (_sc_params : IRobotViewScreenCaptureParams)
Function makes a screen capture of a graphical view.

Panel cuts

Available since version 3.

IRobotViewPanelCuts

Derived from IRobotView3
View presenting cuts through panels.
Available since version 3.

IRobotViewPanelCutParams

Parameters navigating panel cut display.
Available since version 3.

Properties:

CurrentResult : IRobotViewFeMapResultType
Currently displayed result type

Available since version 3.

CurrentReinforcementResult : IRobotViewReinforcementResultType
Currently displayed result type for reinforcement.
Available since version 3.

Descritpions : IRobotViewDiagramDescriptionType
Method of description display
Available since version 3.

IntegralValue : bool
activation of a label containing the value of integral for the selected component along the cutting line length
Available since version 3.

Filling : IRobotViewDiagramFillingType
method of diagram filling
Available since version 3.

Position : IRobotViewDiagramPositionType
type of diagram position with respect to the panel plane
Available since version 3.

Smoothing : IRobotViewFeMapSmoothingType
Type of diagram smoothing
Available since version 3.

Layer : IRobotViewFeMapLayerType
Layer
Available since version 3.

LayerArbritraryValue : double
Layer thickness
Available since version 3.

Influence lines

Available since version 3.5.

IRobotViewInfluenceLines

Derived from IRobotView2
Graphical view presenting an influence line.
Available since version 3.5.

Properties:

ParamsInfluenceLines :

view parameters
Available since version 3.5.

Operations:

MakeScreenCapture (_sc_params : IRobotViewScreenCaptureParams)

IRobotViewInfluenceLinesParams

Parameters of an influence line view.
Available since version 3.5.

Properties:

RangeFrom : long
initial number of the moving load position with which generation of an influence line will start
Available since version 3.5.

RangeTo : long
final number of the moving load position (associated with RangeFrom)
Available since version 3.5.

Element : long
number of element/bar for which an influence line will be generated
Available since version 3.5.

Position : double
relative coordinate of the point of element/bar for which an influence line will be generated

Available since version 3.5.

DirectionData : IRobotGeoPoint3D
vector defining the local reference system; used in appropriate system types (CARTESIAN_ALONG_VECTOR and POLAR_IN_POINT).
Available since version 3.5.

DirectionNode : long
node number for definition of the POLAR_IN_POINT local reference system
Available since version 3.5.

LayerArbitraryVal : double
value of the ARBITRARY type layer
Available since version 3.5.

Layer : IRobotViewInfluenceLinesLayerType
layer type
Available since version 3.5.

Direction : IRobotViewInfluenceLinesLocalSystemType
type of the local reference system
Available since version 3.5.

Operations:

Set (_type : IRobotViewInfluenceLinesResultTypes, _val : bool)
Function manages display of results of the specified type.
Available since version 3.5.

IsOn (_type : IRobotViewInfluenceLinesResultTypes) : bool
Function checks if a given result type is currently displayed.
Available since version 3.5.

IRobotViewInfluenceLinesResultType

Available result types for the influence line view.
Available since version 3.5.

Properties:

I_VILRT_NTM_FX : = 1

Available since version 3.5.

I_VILRT_NTM_FY : = 2

Available since version 3.5.

I_VILRT_NTM_FZ : = 3

Available since version 3.5.

I_VILRT_NTM_MX : = 4

Available since version 3.5.

I_VILRT_NTM_MY : = 5

Available since version 3.5.

I_VILRT_NTM_MZ : = 6

Available since version 3.5.

I_VILRT_NTM_KY : = 7

Available since version 3.5.

I_VILRT_NTM_KZ : = 8

Available since version 3.5.

I_VILRT_NTM_UX : = 9

Available since version 3.5.

I_VILRT_NTM_UZ : = 10

Available since version 3.5.

I_VILRT_NODES_UX : = 11

Available since version 3.5.

I_VILRT_NODES_UY : = 12

Available since version 3.5.

I_VILRT_NODES_UZ : = 13

Available since version 3.5.

I_VILRT_NODES_RX : = 14

Available since version 3.5.

I_VILRT_NODES_RY : = 15

Available since version 3.5.

I_VILRT_NODES_RZ : = 16

Available since version 3.5.

I_VILRT_NODES_FX : = 17

Available since version 3.5.

I_VILRT_NODES_FY : = 18

Available since version 3.5.

I_VILRT_NODES_FZ : = 19

Available since version 3.5.

I_VILRT_NODES_MX : = 20

Available since version 3.5.

I_VILRT_NODES_MY : = 21

Available since version 3.5.

I_VILRT_NODES_MZ : = 22

Available since version 3.5.

I_VILRT_DETAILED_STRESS_XX : = 23

Available since version 3.5.

I_VILRT_DETAILED_STRESS_YY : = 24

Available since version 3.5.

I_VILRT_DETAILED_STRESS_XY : = 25

Available since version 3.5.

I_VILRT_DETAILED_STRESS_Z : = 26

Available since version 3.5.

I_VILRT_DETAILED_MEMBRANE_FORCE_XX : = 27

Available since version 3.5.

I_VILRT_DETAILED_MEMBRANE_FORCE_YY : = 28

Available since version 3.5.

I_VILRT_DETAILED_MEMBRANE_FORCE_XY : = 29

Available since version 3.5.

I_VILRT_DETAILED_MOMENTS_XX : = 30

Available since version 3.5.

I_VILRT_DETAILED_MOMENTS_YY : = 31

Available since version 3.5.

I_VILRT_DETAILED_MOMENTS_XY : = 32

Available since version 3.5.

I_VILRT_DETAILED_SHEAR_STRESS_XX : = 33
I_VILRT_DETAILED_SHEAR_STRESS_YY : = 34

Available since version 3.5.

I_VILRT_DETAILED_SHEAR_FORCE_XX : = 35

Available since version 3.5.

I_VILRT_DETAILED_SHEAR_FORCE_YY : = 36

Available since version 3.5.

I_VILRT_DETAILED_DISPLACEMENT_XX : = 37

Available since version 3.5.

I_VILRT_DETAILED_DISPLACEMENT_YY : = 38

Available since version 3.5.

I_VILRT_DETAILED_DISPLACEMENT_Z : = 39

Available since version 3.5.

I_VILRT_DETAILED_ROTATION_XX : = 40

Available since version 3.5.

I_VILRT_DETAILED_ROTATION_YY : = 41

Available since version 3.5.

I_VILRT_DETAILED_ROTATION_Z : = 42

Available since version 3.5.

I_VILRT_DETAILED_SOIL_REACTION_Z : = 43

Available since version 3.5.

I_VILRT_EXTREME_STRESS_1 : = 44

Available since version 3.5.

I_VILRT_EXTREME_STRESS_2 : = 45

Available since version 3.5.

I_VILRT_EXTREME_STRESS_1_2 : = 46

Available since version 3.5.

I_VILRT_EXTREME_STRESS_ANGLE : = 47

Available since version 3.5.

I_VILRT_EXTREME_MEMBRANE_FORCES_1 : = 48

Available since version 3.5.

I_VILRT_EXTREME_MEMBRANE_FORCES_2 : = 49

Available since version 3.5.

I_VILRT_EXTREME_MEMBRANE_FORCES_1_2 : = 50

Available since version 3.5.

I_VILRT_EXTREME_MEMBRANE_FORCES_ANGLE : = 51

Available since version 3.5.

I_VILRT_EXTREME_MOMENTS_1 : = 52

Available since version 3.5.

I_VILRT_EXTREME_MOMENTS_2 : = 53

Available since version 3.5.

I_VILRT_EXTREME_MOMENTS_1_2 : = 54

Available since version 3.5.

I_VILRT_EXTREME_MOMENTS_ANGLE : = 55

Available since version 3.5.

I_VILRT_EXTREME_SHEAR_STRESS_1_2 : = 56

Available since version 3.5.

I_VILRT_EXTREME_SHEAR_FORCE_1_2 : = 57

Available since version 3.5.

I_VILRT_COMPLEX_STRESS : = 58

Available since version 3.5.

I_VILRT_COMPLEX_MEMBRANE_FORCE : = 59

Available since version 3.5.

I_VILRT_COMPLEX_MOMENT : = 60

Available since version 3.5.

IRobotViewInfluenceLinesLocalSystemType

Available types of local coordinate systems.
Available since version 3.5.

Properties:

I_VILLST_CARTESIAN_ALONG_X : = 0

Available since version 3.5.

I_VILLST_CARTESIAN_ALONG_Y : = 1
I_VILLST_CARTESIAN_ALONG_Z : = 2

Available since version 3.5.

I_VILLST_CARTESIAN_ALONG_VECTOR : = 3

Available since version 3.5.

I_VILLST_POLAR_IN_NODE : = 4

Available since version 3.5.

I_VILLST_POLAR_IN_POINT : = 5

Available since version 3.5.

I_VILLST_AUTOMATIC : = 6

Available since version 3.5.

IRobotViewInfluenceLinesLayerType

Types of layers.
Available since version 3.5.

Properties:

I_VILLT_UPPER : = 0

Available since version 3.5.

I_VILLT_MIDDLE : = 1

Available since version 3.5.

I_VILLT_LOWER : = 2

Available since version 3.5.

I_VILLT_ARBITRARY : = 3

Available since version 3.5.

IRobotViewDiagramValueType

Types of values displayed on diagrams.
Available since version 8.1.

Properties:

I_VDVT_ALL : = 0

Available since version 8.1.

I_VDVT_LOCAL_EXTREMES : = 1

Available since version 8.1.

I_VDVT_GLOBAL_EXTREMES : = 2

Available since version 8.1.

Tables

IRobotTableType

One can differentiate a number of table types presenting different types of data. Each table type has got its identifier.

IRobotTableDataType

A set of identifiers is defined to refer to types of data presented in tables.

IRobotTableConfigFlag

Tables may be configured in many ways. A set of flags is defined to help one configure tables.

Properties:

I_TCF_DETAILS : = 1
detailed information
Available since version 1.7.

I_TCF_GLOBAL_COORDINATES : = 2
global coordinates
Available since version 1.7.

I_TCF_LOCAL_COORDINATES : = 3
local coordinates
Available since version 1.7.

I_TCF_COEXISTENT_VALUES : = 4
Available since version 1.7.

I_TCF_EXTREME_COMBINATIONS : = 5
Available since version 1.7.

I_TCF_PROP_STD_TAPERED_SECTION : = 6
Available since version 1.7.

I_TCF_COMB_DEFINITIONS : = 7
Available since version 1.7.

I_TCF_COMB_CODE_COMPONENTS : = 8
Available since version 1.7.

I_TCF_COMB_MOVING_LOADS : = 9
Available since version 1.7.

I_TCF_COMB_TIME_HISTORY_COMPONENTS : = 10
Available since version 1.7.

I_TCF_REAC_VALUES : = 11
Available since version 1.7.

I_TCF_REAC_APPLIED_FORCES_SUM : = 12
Available since version 1.7.

I_TCF_REAC_EQUILIBRIUM_PRECISION : = 13
Available since version 1.7.

I_TCF_REAC_RESIDUUM : = 14
Available since version 1.7.

I_TCF_BAR_DEFL_DEFLECTIONS : = 15
Available since version 1.7.

I_TCF_BAR_DEFL_MAX_DEFLECTIONS : = 16
Available since version 1.7.

I_TCF_BAR_DEFL_DISPLACEMENTS : = 17
Available since version 1.7.

I_TCF_FE_DIR_X_AXIS : = 18
Available since version 1.7.

I_TCF_FE_DIR_Y_AXIS : = 19
Available since version 1.7.

I_TCF_FE_DIR_Z_AXIS : = 20
Available since version 1.7.

I_TCF_FE_LAYER_UPPER : = 21
Available since version 1.7.

I_TCF_FE_LAYER_MIDDLE : = 22
Available since version 1.7.

I_TCF_FE_LAYER_LOWER : = 23
Available since version 1.7.

I_TCF_FE_LAYER_MAX : = 24
Available since version 1.7.

I_TCF_FE_LAYER_MIN : = 25
Available since version 1.7.

I_TCF_FE_LAYER_MAX_ABSOLUTE : = 26
Available since version 1.7.

I_TCF_FE_IN_NODES : = 27
Available since version 1.7.

I_TCF_FE_IN_ELEMENT_CENTERS : = 28
Available since version 1.7.

I_TCF_REINF_IN_NODES : = 29
Available since version 1.7.

I_TCF_REINF_IN_ELEMENT_CENTERS : = 30
Available since version 1.7.

I_TCF_REAC_DDC : = 31
Available since version 1.7.

I_TCF_LOCALS : = 32
Available since version 1.7.

I_TCF_FE_IN_ELENODE : = 33
Available since version 1.7.

I_TCF_ORDER_BY_CASE : = 34
Available since version 1.7.

I_TCF_CASE_INFO : = 35
Available since version 1.7.

I_TCF_GLOBAL_ANALYSIS_RESULTS_NPOINTS : = 36

Available since version 3.

I_TCF_GLOBAL_ANALYSIS_RESULTS_RELATIVE : = 37

Available since version 3.

IRobotTableConfigValue

Set of values used to configure tables.

IRobotTableConfig

Each table may be configured by means of a configuring object.

Operations:

SetFlag (flag : IRobotTableConfigFlag, set : bool)
The function switches {on/off} the indicated flag configuring the table.

SetValue (value_id : IRobotTableConfigValue, val : double)
The function sets the value of the indicated variable configuring the table.

IRobotTable

Interface representing tables.

Properties:

Visible : bool
variable determining if the table is to be displayed or not

Title : string
table title

Printable : IRobotPrintable
object representing tables; it may be added to the printout and printed

Configuration : IRobotTableConfig
object controlling table configuration

Window : IRobotWindow

Available since version 3.

UserControl : bool
flag that enables giving/taking back table navigation to/from the interactive program user

Available since version 3.

Operations:

AddColumn (data_type : IRobotTableDataType)
The function adds to a table a new column that will present data of the indicated type

Select (sel_type : IRobotObjectType, sel_text : string)
The function sets a filter defined by the indicated selection on the table

StoreSelection (sel_type : IRobotObjectType, sel_txt : string)
The function saves the indicated selection in the table without modifying its filter. One may save one selection of each type of objects - structure components - in the table.

SelectFromStore (sel_type : IRobotObjectType)
The function sets a filter on the table. The filter is defined by means of the selection saved earlier by the StoreSelection() function.

SelectAllFromStore ()
The function sets a filter on the table, defined by means of all selections formerly saved in the table by means of the StoreSelection() function.

SelectMode (type : IRobotModeSelectionType, sel_text : string, comb : IRobotModeCombinationType)
The function sets a filter on the table, described by the indicated mode selection.

StoreModeSelection (type : IRobotModeSelectionType, sel_text : string, comb : IRobotModeCombinationType)
The function saves the indicated mode selection to the table.

SelectModeFromStore ()
The function sets a filter on the table, defined by means of mode selection formerly saved in the table by means of the StoreModeSelection().

IRobotTableFrame

Interface representing the viewer composed of many tables. Each table is a separate tab defined within the same main frame.
Available since version 3.

Properties:

Count : long
number of tables defined in the same main frame
Available since version 3.

Window : IRobotWindow

Available since version 3.

Current : long
Index of active table
Available since version 3.

Operations:

Get (_idx : long) : IRobotTable
Function returns the table of the specified index.
Available since version 3.

GetName (_table_idx : long) : string
Function returns the name of the tab on which the table of the indicated index is displayed.
Available since version 3.

SetName (_table_idx : long, _name : string)
Function sets the name for the tab which presents the table of the indicated index.
Available since version 3.

Viewers for RTF files

IRobotRtfView

Files in RTF format are frequently used to present data or results. Robot enriched the format with the possibility of introducing into an RTF file the variables defined in the program. A variable is written to file in the form of its name, marked with the "@" sign at both ends. The view RobotRtfView automatically extends variables to give them their current value.

Properties:

Printable : IRobotPrintable
object representing a view, that may be added to the print-out and printed

Operations:

LoadFromFile (file_path : string) : bool
The function reads the indicated file in RTF format.

SaveToFile (file_path : string, as_template : bool = false) : bool
The function saves the content of the view to the indicated file. If the "as_template" parameter is not equal zero (set to True), the view contents will be written as a template (variable names and not their values will be saved to file).

AppendFromFile (file_path : string) : bool
The function pastes the contents of the indicated view at the end of the view.

Evaluate ()
The function extends all the variables in the text and changes their names to their current values.
There is no corresponding undo operation.

Available since version 7.5.

IRobotHtmlView

Available since version 7.5.

Properties:

Printable : IRobotPrintable
object representing a view that may be attached to the printout and printed
Available since version 7.5.

Operations:

LoadFromFile (_file_path : string) : bool
Function loads an HTML format file.
Available since version 7.5.

SaveToFile (_file_path : string) : bool
Function saves the view contents to the specified HTML format file.
Available since version 7.5.

Screen captures

The Robot program allows saving the current state (settings and/or contents) of a selected view into the archive to be used later at the stage of creating documentation. It is performed by making a screen capture of a selected view. The screen captures are identified by their names.

IRobotScreenCaptureMngr

An object manages screen captures defined in the current project.

Properties:

Count : long
number of defined screen captures
Available since version 1.7.

Operations:

Get (idx : long) : string
Function returns name of a screen capture with a specified index. Screen captures are indexed from 1 to Count.
Available since version 1.7.

Remove (sc_name : string)
Function deletes the archive containing a screen capture with a specified name from the project.
Available since version 1.7.

Find (sc_name : string) : long
Function returns index of a screen capture with a specified name. If screen capture with this name does not exist, then zero value is returned.
Available since version 1.7.

Printouts

IRobotPrintable

Interface of any data object that is to be printed by the printing module of Robot.

Properties:

Title : string
title

StartFromNewPage :
falg indicating whether an object should be printed from a new page

Operations:

SaveToFile (file_path : string, file_format : IRobotOutputFileFormat) : bool
Function exports object contents to the indicated file with the specified format.
Available since version 1.7.

IRobotPrintEngine

Interface representing the Robot printing module.

Properties:

PageSetup : IRobotPageSetup
page setup

ScreenCaptures : IRobotScreenCaptureMngr
list of available screen captures
Available since version 1.7.

ReportTemplates : IRobotReportTemplateMngr
list of available printout templates
Available since version 1.7.

OrganizerItems : IRobotReportItemList
list of printout components defined in the Printout Composition dialog box
Available since version 5.5.

IsWhilePreview : bool
flag indicating that the application is currently working in the "pint preview" mode

Operations:

ResetReport ()
The function deletes all objects from the current printout.

AddToReport (obj : IRobotPrintable)

The function adds the indicated object at the end of the currently compoesed printout.

RemoveFromReport (obj : IRobotPrintable)
The function deletes the indicated object from the current printout.

PreviewReport ()
The function runs the print preview of the current printout composition

PrintReport ()
The function results in sending the current printout composition to the printer.

SaveReportToFile (file_path : string, file_format : IRobotOutputFileFormat) : bool
Function exports printout to the specified file with the defined format.
Available since version 1.7.

AddTemplateToReport (tmpl_name : string) : bool
Function adds contents of a printout template with a specified name at the end of the printout. If a template with this name is not found, then function returns zero value (False).
Available since version 1.7.

SaveReportToTemplate (tmpl_name : string)
Function saves printout to the template with a specified name. If a template with this name already exists its contents will be replaced.
Available since version 1.7.

AddScToReport (sc_name : string) : bool
Function adds screen capture with a specified name at the end of the current printout. If a screen capture with a specified name is not found, zero value is returned (False).
Available since version 1.7.

RemoveScFromReport (sc_name : string)
Function deletes screen capture with a specified name from the current printout. First screen capture with this name occurring on the printout component list is deleted.
Available since version 1.7.

CreateReportFromOrganizer ()
Function creates a printout composed of elements currently defined in the Printout Composition dialog box.
Available since version 6.4.

SaveReportToOrganizer ()
Function saves the current printout as a current list of printout elements presented in the Printout Composition dialog box.
Available since version 6.4.

ExternalPreviewReport (_format : IRobotExternalPreviewFormat)
Function launches an external program and exports the current printout to it. Available since version 6.4.

ClosePreview ()
Function forces closing a print preview.

IRobotVariableMngr

Manager of variables (parameters) used in printout templates.

Operations:

Exist (var_name : string) : bool
The function checks if there exists a variable with the indicated name

GetValue (var_name : string) : string
The function returns the value of the indicated variable. If the variable with the indicated name does not exist, the function returns an empty string.

SetValue (var_name : string, var_value : string)
The function ascribes a defined value to the variable with the given name. If the variable does not exist, it will be defined. If it does exist, its value will be changed.

Delete (var_name : string)
The function deletes a variable with the given name.

GetPredefinedValue (var_id : IRobotVariablePredefinedId) : string
Function returns the value of the given standard variable.

SetPredefinedValue (var_id : IRobotVariablePredefinedId, var_value : string) : bool
Function sets value of a specified standard variable. Some variables are read-only. For such a variable, values cannot be set. Then, this function returns zero (False).

AddExtension (_ext_obj : IRobotVariableMngrExtension)
Function enables extending the manager with an external object providing values for variables.

RemoveExtension (_ext_obj : IRobotVariableMngrExtension)
Function deletes a specified object from the extension list.

IRobotVariablePredefinedId

Set of identifiers for standard (previously prepared for the user) variables.

Properties:

I_VPI_REPORT_PAGE_NUMBER : = 0
number of a print-out page (initial or current during printing)

I_VPI_DATE : = 1
current date

I_VPI_TIME : = 2
current time

I_VPI_ROBOT_APP : = 3
program name

I_VPI_ROBOT_VER : = 4
program version

I_VPI_ROBOT_PROVIDER : = 5
name of the program supplier (dealer)

I_VPI_USER_NAME : = 6
user's name

I_VPI_USER_ADDRESS : = 7
user's address

I_VPI_PROJECT_NAME : = 8
project name

I_VPI_PROJECT_FILE : = 9
name of the file containing the project

I_VPI_PROJECT_DIRECTORY : = 10
folder where the file with the project is stored

I_VPI_PROJECT_SIZE : = 11
size of the file with the project

I_VPI_PROJECT_CREATED : = 12
date when the project file was created

I_VPI_PROJECT_MODIFIED : = 13
date of the last modification of the project file

I_VPI_PROJECT_VERSION : = 14
project version

I_VPI_PROJECT_ADDRESS1 : = 15
I_VPI_PROJECT_ADDRESS2 : = 16
I_VPI_PROJECT_REFERENCE : = 17
I_VPI_PROJECT_STEP : = 18
I_VPI_PROJECT_COMMENT : = 19
I_VPI_ARCHITECT_NAME : = 20
I_VPI_ARCHITECT_ADDRESS1 : = 21
I_VPI_ARCHITECT_ADDRESS2 : = 22

I_VPI_ARCHITECT_CONTACT : = 23
I_VPI_ARCHITECT_TELEPHONE : = 24
I_VPI_ARCHITECT_FAX : = 25
I_VPI_ARCHITECT_EMAIL : = 26
I_VPI_INVESTOR_NAME : = 27
I_VPI_INVESTOR_ADDRESS1 : = 28
I_VPI_INVESTOR_ADDRESS2 : = 29
I_VPI_INVESTOR_CONTACT : = 30
I_VPI_INVESTOR_TELEPHONE : = 31
I_VPI_INVESTOR_FAX : = 32
I_VPI_INVESTOR_EMAIL : = 33
I_VPI_COMPANY_NAME : = 34
I_VPI_COMPANY_ADDRESS1 : = 35
I_VPI_COMPANY_ADDRESS2 : = 36
I_VPI_COMPANY_CONTACT : = 37
I_VPI_COMPANY_TELEPHONE : = 38
I_VPI_COMPANY_FAX : = 39
I_VPI_COMPANY_EMAIL : = 40
I_VPI_DESIGNER_NAME : = 41
I_VPI_DESIGNER_ADDRESS1 : = 42
I_VPI_DESIGNER_ADDRESS2 : = 43
I_VPI_DESIGNER_CONTACT : = 44
I_VPI_DESIGNER_TELEPHONE : = 45
I_VPI_DESIGNER_FAX : = 46
I_VPI_DESIGNER_EMAIL : = 47
I_VPI_VERIF_NAME : = 48
I_VPI_VERIF_ADDRESS1 : = 49
I_VPI_VERIF_ADDRESS2 : = 50
I_VPI_VERIF_CONTACT : = 51
I_VPI_VERIF_TELEPHONE : = 52
I_VPI_VERIF_FAX : = 53
I_VPI_VERIF_EMAIL : = 54
I_VPI_PROJECT_STATISTICS : = 55
I_VPI_PROJECT_CONTENT : = 56

IRobotPageSetup

Interface allows reading and modifying current page settings and enables template management. The Robot program allows definition of many templates storing page settings, which are differentiated by name. At the given moment one of the templates is set as current (default).

Properties:

Variables : IRobotVariableMngr
manager of variables (parameters)

TitlePage : IRobotReportStdElementRtf
printout title page
Available since version 2.5.

Header : IRobotReportStdElementRtf
header on each printout page
Available since version 2.5.

Footer : IRobotReportStdElementRtf
footer on each printout page

Available since version 2.5.

PaperSize : long
identifier of paper size
Available since version 5.5.

PaperWidth : long
paper width in millimeters
Available since version 5.5.

PaperHeight : long
paper height in millimeters
Available since version 5.5.

StartPageNumber : long
initial page number
Attention:
Initial page number is not stored in a template. Setting a value of the initial page number means the same as setting a value of the variable named PAGE_NUMBER by means of the IRobotVariableMngr interface.
Available since version 5.5.

PageOrientation : IRobotPageSetupOrientation
page setup
Available since version 5.5.

MarginTop : double
top margin in millimeters
Available since version 5.5.

MarginLeft : double
left margin in millimeters
Available since version 5.5.

MarginBottom : double
bottom margin in millimeters
Available since version 5.5.

MarginRight : double
right margin in millimeters
Available since version 5.5.

Gutter : double
gutter in millimeters

Available since version 5.5.

FromEdgeHeader : double
distance of header from the top edge of paper in millimeters
Available since version 5.5.

FromEdgeFooter : double
distance of footer from the bottom edge of paper in millimeters
Available since version 5.5.

TextFrame : IRobotPageSetupFrameType
page contents frame
Available since version 5.5.

Toc : IRobotPageSetupTableOfContents
parameters of the table of contents
Available since version 5.5.

TemplateName : string
template name
Available since version 5.5.

IsCurrent : bool
flag that enables setting page parameters as the current ones
Available since version 5.5.

TemplateCount : long
number of the remembered templates with page settings
Available since version 5.5.

Operations:

SaveAs (_template_name : string) : bool
Function saves current page settings as a template with the specified name.
Available since version 5.5.

Load (_template_name : string) : bool
Function reads page settings from the specified template.
Available since version 5.5.

Save () : bool
Function saves current page settings.
Available since version 5.5.

GetTemplateName (_idx : long) : string
Function returns the name of a template with the specified index. Templates are indexed with values from 1 to TemplateCount.
Available since version 5.5.

LoadCurrent () : bool
Function reads current page settings.

Available since version 5.5.

IRobotOutputFileFormat

Available file formats to which a printout may be directed.

Properties:

I_OFF_TEXT : = 1
Available since version 1.7.

I_OFF_RTF : = 2
Available since version 1.7.

I_OFF_RTF_JPEG : = 3
RTF format file with the inserted JPEG format graphics
Available since version 5.5.

I_OFF_HTML : = 4
HTML format file
Available since version 7.5.

IRobotReportTemplateMngr

There is a possibility to save a previously composed printout as a template, which may be used many times afterwards in the same or different project. Printout templates are recognized by names.
All templates available in the current project are managed by the template manager.

Properties:

Count : long
number of available printout templates
Available since version 1.7.

Operations:

Get (idx : long) : string
Function returns name of a template with a specified index. Printout templates are indexed from 1 to Count.
Available since version 1.7.

Remove (tmpl_name : string)
Function deletes a printout template with the specified name.
Available since version 1.7.

Find (tmpl_name : string) : long
Function finds a printout template with the specified name and returns its index. If the printout template is not found, then zero value is returned.

Available since version 1.7.

IRobotReportStdElementRtf

Standard (predefined) printout element based on an RTF format template file.
Available since version 2.5.

Properties:

Active : bool
flag indicating if the element is active (i.e. it will be added to the printout)
Available since version 2.5.

Frame : IRobotPageSetupFrameType
frame type
Available since version 5.5.

Operations:

LoadFromFile (_file_path : string) : bool
Function reads in the printout element contents from the indicated file.
Available since version 2.5.

RestoreDefaults ()
Function restores the standard contents of a printout element.
Available since version 2.5.

SaveToFile (_fpath : string, _format : IRobotOutputFileFormat) : bool
Function saves an element to a file of selected format.
Available since version 5.5.

IRobotReportItemList

List of printout components.
Available since version 5.5.

Properties:

Count : long
number of elements on the list
Available since version 5.5.

Operations:

Get (_idx : long) : IRobotReportItem
Function returns a printout component with the specified index. Printout components are indexed from 1 to Count.
Available since version 5.5.

IRobotReportItem

Interface that describes a printout element.
Available since version 5.5.

Properties:

StartFromNewPage : bool
flag that indicates if an element should be printed on a new page
Available since version 5.5.

TitleText : string
element title
Available since version 5.5.

Title : IRobotRtfView
element title in RTF format
Available since version 5.5.

HasNoteBefore : bool
flag indicating that a note has been inserted before the printout component
Available since version 5.5.

NoteBefore : IRobotRtfView
note inserted before the printout component
Available since version 5.5.

HasNoteAfter : bool
flag indicating that at a note has been inserted after the printout component
Available since version 5.5.

NoteAfter : IRobotRtfView
note inserted after the printout component

Available since version 5.5.

Type : IRobotReportItemType
type of printout element
Available since version 5.5.

Operations:

CreateView () : IDispatch*
Fuction creates and returns a view corresponding to the printout component. A type of the created view depends on the contents of the printout component. If access to any of the defined interfaces for views is impossible, then the IRobotPrintable interface is returned.
In case an error occurs, the empty reference is returned (zero for the C language, Nothing for Visual Basic).

Function makes the interface accessible for the object reperesented by the printout component of I_RIT_VIEW type.
Available since version 5.5.

GetPageTemplate () : string
Function returns the name of a page template for the printout component of I_RIT_PAGE_TEMPLATE type.
Available since version 5.5.

IRobotPageSetupOrientation

Page setup.
Available since version 5.5.

Properties:

I_PSO_PORTRAIT : = 1

Available since version 5.5.

I_PSO_LANDSCAPE : = 2

Available since version 5.5.

IRobotPageSetupFrameType

Types of frames used for decoration of individual page elements.
Available since version 5.5.

Properties:

I_PSFT_NONE : = 0
no frame
Available since version 5.5.

I_PSFT_FRAME : = 1
frame
Available since version 5.5.

I_PSFT_SEPARATION_LINE : = 2
separation line
Available since version 5.5.

IRobotPageSetupTableOfContents

Interface that describes the table of contents.
Available since version 5.5.

Properties:

Active : bool
flag that switches on/off the table of contents
Available since version 5.5.

Location : IRobotPageSetupTocLocation

Available since version 5.5.

IncludeTitle : bool

Available since version 5.5.

IRobotPageSetupTocLocation

Available since version 5.5.

Properties:

I_PSTL_BEGINNING : = 1
at the beginning
Available since version 5.5.

I_PSTL_END : = 2
at the end
Available since version 5.5.

IRobotReportItemType

Available types of printout components.
Available since version 5.5.

Properties:

I_RIT_BLANK_PAGE : = 1
printout component that represents a blank page
Available since version 5.5.

I_RIT_PAGE_TEMPLATE : = 2
printout component that enforces change of the page template
Available since version 5.5.

I_RIT_VIEW : = 3
printout component representing a view or other object that may be printed
Available since version 5.5.

IRobotExternalPreviewFormat

External format of a print preview.
Available since version 6.4.

Properties:

EPF_HTML : = 1
print preview in the html file browser
Available since version 6.4.

EPF_MS_OFFICE : = 2
presentation of a printout in the Word program of the Microsoft Office package; it requires installing the Microsoft Office package
Available since version 6.4.

EPF_OPEN_OFFICE : = 3
presentation of a printout in the text editor of the Open Office package; it requires installing the Open Office package
Available since version 6.4.

IRobotVariableMngrExtension

Interface of external extension for the manager of variables (parameters) used in printouts.

Operations:

GetValue (_var_name : string, _var_value : string*) : bool
Function fills out a value of the variable of a given name. If the object does not support this variable, then function returns FALSE value (zero).

GetIndexedValue (_var_name : string, _cur_index : long, _max_index : long, _var_value : string*) : bool
Function returns a value of the variable indexed for a given value of an index. If the object does not support the variable of a given name, then function returns FALSE value (zero).

Add-in manager

IRobotAddIn

Definition of the Robot program extension.
This interface must implement each component that is integrated with the Robot program and that extends its functionality.

Operations:

GetExpectedVersion () : double

Function should return the number of the RobotOS model version which is used by the extension.
It will enable warning a user in a situation when the extension that is being installed by the user expects a newer Robot program version than the one currently used by the user.

Connect (robot_app : IRobotApplication, add_in_id : long, first_time : bool) : bool

Function wil be called up by the Robot application when the extension is connected to the program: for the first time - upon user's request or during first activation of the program after installing the extension; for the next time - while activating the Robot program again.
The extension should remember the access interface to the Robot application (RobotApplication) so that the extension can use its functionality.
The extension, while being connected, is ascribed its identifier which should be remembered since it will be needed during subsequent communication with the Robot program.
While connecting the extension to the application for the first time it is possible to perform additional initializing operations.
If all is performed correctly, then function should return a value different from zero (True).

InstallCommands (cmd_list : IRobotCmdList) : bool

Function will be called up by the application to obtain information about the commands that are made available by the extension.
List specified by the parameter should be filled out with the information about all the commands defined by the extension.
If the operation is performed successfully, the function should return a value different from zero (True).

Disconnect () : bool

This function will be called up before the extension is disconnected with the Robot program (e.g. while exiting the application or upon user's request who wants no longer to use the extension). If there is any reason for the extension not to be disconnected with the program at the given moment, the function should return zero value (False).

DoCommand (cmd_id : long)

Function will be called up by the Robot program if a user selects the menu option linked with the command supported by this extension.

IRobotAddInMngr

Add-ins Manager keeps tracks to all add-ins currently registered in Robot.

Operations:

InstallCommand (add_in_id : long, cmd_id : long, cmd_name : string) : bool

IRobotCmdList

List of information about commands.

Properties:

Count : long
number of elements of the list
Available since version 1.7.

Operations:

New (cmd_id : long, cmd_name : string) : long
Function creates a new list element and adds it at the end. The index of the element on the list is returned.
Available since version 1.7.

Get (idx : long) : IRobotCmdInfo
Function returns the specified list element. List elements are indexed from 1 to Count.
Available since version 1.7.

IRobotCmdInfo

Information about the command and the menu option linked with it.

Properties:

Id : long
command identifier
Available since version 1.7.

Name : string
name of the menu option linked with the command
Available since version 1.7.

MenuEnabled : bool
flag indicating if the menu option linked with the command is available
Available since version 1.7.

MenuChecked : long
indicator of marking the menu option linked with the command
0 - not marked
1 - marked
2 - intermediate state

Available since version 1.7.

IRobotAddInRegistrar

Information about each Robot program extension must be saved as an appropriate position in the system register.
RobotAddlnRegistrar facilitates the process of extension registration.

Properties:

ProviderName : string
name of the extension supplier
Available since version 1.7.

ProductName : string
extension name
Available since version 1.7.

Guid : string
GUID identifier of the main extension component (presented interchangeably with Progld)
Available since version 1.7.

ProgId : string
identifier of the main extension component (presented interchangeably with Guid)
Available since version 1.7.

Operations:

Register () : bool
Function saves information about the extension to the register. If registration is performed successfully, a value different from zero is returned (True).
Registration failure may result from lack of settings concerning suppler name or product as well as identifier (Guid or Progld).
Available since version 1.7.

Unregister ()
Function deletes information about the extension from the register.
Available since version 1.7.

InstallMenu (menu_name : string, options : IRobotCmdList) : bool
Function adds menu with the specified name to all Robot program views. The commands specified in the list will are the menu options. They will be supported by the extension whose Progld or Guid have been set earlier. If identifier of the extension supporting commands from the list is not specified, the function returns zero value (False).
Available since version 1.7.

Concrete

IRConcrBarSectionData

A simplified interface describing an RC member section.

Properties:

N : int
H : double
B : double
Bf : double
Hf : double
Bfb : double
Hfb : double
A : double
Ap : double
Ac : double
Per : double
Type : RConcrBarSectionGeometryType
section geometry type

IRConcrBarSectionGeometryType

Types of geometry of an RC member section

Properties:

I_CBSGT_R : = 0
I_CBSGT_L : = 1
I_CBSGT_T : = 2
I_CBSGT_TN : = 3
I_CBSGT_I : = 4
I_CBSGT_Z : = 5
I_CBSGT_P : = 6
I_CBSGT_C : = 7
I_CBSGT_C2 : = 8
I_CBSGT_C4 : = 9
I_CBSGT_UR : = 10
I_CBSGT_UL : = 11
I_CBSGT_UT : = 12
I_CBSGT_UZ : = 13
I_CBSGT_UP : = 14
I_CBSGT_UC : = 15
I_CBSGT_UC2 : = 16
I_CBSGT_UC4 : = 17

IRConcrCalcEngine

RC calculation module.
Available since version 5.5.

Properties:

SlabRequiredReinf : RConcrSlabRequiredReinfEngine
module for calculation of the required (theoretical) reinforcement for plates and shells
Available since version 5.5.

MemberRequiredReinf : RConcrMemberRequiredReinfEngine
module calculating the required reinforcement for RC members
Available since version 7.5.

Plate and shell reinforcement

IRConcrReinforceData

Definition of plate and shell reinforcement type includes parameters that are code-independent.

Properties:

BarDim_D1_Bot : double
diameter of reinforcing bars of bottom main reinforcement
Available since version 1.7.

BarDim_D1_Up : double
diameter of reinforcing bars of top main reinforcement
Available since version 1.7.

BarDim_D2_Bot : double
reinforcing bar diameter of bottom reinforcement in the direction perpendicular to the main reinforcement
Available since version 1.7.

BarDim_D2_Up : double
reinforcing bar diameter of top reinforcement in the direction perpendicular to the main reinforcement
Available since version 1.7.

Cover_Bot : double
cover - bottom reinforcement
Available since version 1.7.

Cover_Up : double
cover - top reinforcement
Available since version 1.7.

CodeName : string
code name

Available since version 1.7.

ReinforcingSteel : RConcrSteelParams
Available since version 1.7.

Concrete : RConcrConcreteParams
Available since version 1.7.

Operations:

GetMainDirection (_x : double*, _y : double*, _z : double*) : RConcrReinforceDirection
Function returns the main reiforcement direction. Interpretation of returned coordinate values depends on the manner of determinig direction (direction vector coordinates, if the direction is determined by means of the vector, ALONG_VECTOR, or the point coordinates in the polar system - POLAR_POINT).
Available since version 1.7.

SetMainDirection (dir_type : RConcrReinforceDirection, _x : double, _y : double, _z : double)
Function defines the main reiforcement direction. Parameter interpretation depends on the manner of determinig direction (direction vector coordinates, if the direction is determined by means of the vector, ALONG_VECTOR, or the point coordinates in the polar system - POLAR_POINT, in the remaining cases the coordinate values are ignored).
Available since version 1.7.

IRConcrReinforceDirection

Available reinforcement directions for plates/shells

Properties:

I_CRD_ALONG_X : = 0
Available since version 1.7.

I_CRD_ALONG_Y : = 1
Available since version 1.7.

I_CRD_ALONG_Z : = 2
Available since version 1.7.

I_CRD_CARTESIAN_ALONG_VECTOR : = 3
direction determined in the Cartesian system by means of the specified vector
Available since version 1.7.

I_CRD_POLAR_POINT : = 4
direction determined in the polar system by means of the specified point
Available since version 1.7.

I_CRD_AUTOMATIC : = 5

automatic direction compatible to local X axis of the panel
Available since version 7.5.

IRConcrConcreteParams

Data characterizing concrete.

Properties:

Fc : double
characteristic compressive strength;
parameter assumes value that equals -1 when the value of the appropriate material assigned to a panel is taken for calculations
Available since version 1.7.

Fcj : double
value that is used only in the French code and denotes Fc (characteristic compressive strength);
parameter assumes value that equals -1 when the value of the appropriate material assigned to a panel is taken for calculations
Available since version 1.7.

Fc_calc : double
design compressive strength (Fc_calc = Fc / gammaC, where gammaC denotes safety factor);
parameter assumes value that equals -1 when the value of the appropriate material assigned to a panel is taken for calculations
Available since version 1.7.

Ft : double
characteristic tensile strength;
parameter assumes value that equals -1 when the value of the appropriate material assigned to a panel is taken for calculations
Available since version 1.7.

Ec : double
elasticity modulus for concrete;
parameter assumes value that equals -1 when the value of the appropriate material assigned to a panel is taken for calculations

Available since version 1.7.

Grade : short
identifier denoting a predefined concrete class (according to the code it should be compared with the appropriate set of defined identifiers)
Available since version 1.7.

IRConcrSteelParams

Set of parameters describing reinforcing steel properties

Properties:

Fy : double
characteristic yield point
Available since version 1.7.

Ft : double
characteristic tensile strength
Available since version 1.7.

Fy_calc : double
calculation yield point(Fy_calc = Fy / gammaS, where gammaS is a safety factor)
Available since version 1.7.

Ft_calc : double
calculation tensile strength (Ft_calc = Ft / gammaS, where gammaS is a safety factor)
Available since version 1.7.

Ey : double
value of limit elastic strain
Available since version 1.7.

Grade : short
identifier denoting a predefined steel grade (according to code it should be compared with the appropriate set of defined identifiers)
Available since version 1.7.

IRConcr_PN84_ReinforceData

Derived from IRConcrReinforceData
Parameters of plate and shell reinforcement for the PN-84 code.
Steel grades are described by the identifier set RConr_PN_SteeGrades, concrete classes by RConcr_PN84_ConcreteGrades, whereas types of exposure rating RConcr_PN84_ExposureRatings.

Properties:

CheckConstructionStages : bool
structure check in the phases of structure construction
Available since version 1.7.

SingleShortLoad : bool
single-time short-term load
Available since version 1.7.

LongActionAbove100 : bool

long-term action of the temperature over 100 degrees
Available since version 1.7.

SLS_Cracking : bool
flag indicating if cracking width is to be calculated
Available since version 1.7.

SLS_Deflection : bool
flag indicating if a deflection value for RC slab taking cracking into account is to be calculated
Available since version 1.7.

SLS_CrackingReinfCorrect : bool
flag switching on automatic correction of cracking width by increasing reinforcement area
Available since version 1.7.

SLS_DeflectionReinfCorrect : bool
flag switching on automatic correction of deflections by increasing reinforcement area
Available since version 1.7.

SLS_Exposure : short
exposure rating
Available since version 1.7.

SLS_MaxCracking : double
admissible cracking width
Available since version 1.7.

SLS_MaxDeflection : double
admissible deflection value
Available since version 1.7.

SLS_LoadsRatio : double
long-term loads to total loads ratio
Available since version 1.7.

SLS_ConcreteAge : short
concrete age (days)
Available since version 1.7.

SLS_EnvHumidity : RConcr_PN84_HumidityType
type of environment humidity
Available since version 1.7.

SLS_CreepingCoef : double
concrete creeping coefficient
Available since version 1.7.

Main : RConcrReinforceDataMain

IRConcr_PN_SteelGrades

Reinforcing steel grades for Polish codes: PN-84 and PN-99

Properties:

I_CPNSG_A_0 : = 2
Available since version 1.7.

I_CPNSG_A_I : = 3
Available since version 1.7.

I_CPNSG_A_II : = 4
Available since version 1.7.

I_CPNSG_A_III : = 5
Available since version 1.7.

I_CPNSG_A_IIIN : = 6
Available since version 1.7.

IRConcr_PN84_ConcreteGrades

Concrete class for Polish code PN-84

Properties:

I_CPN84CG_ANOTHER : = -1
other concrete type
Available since version 1.7.

I_CPN84CG_B10 : = 0
Available since version 1.7.

I_CPN84CG_B12_5 : = 1
Available since version 1.7.

I_CPN84CG_B15 : = 2
Available since version 1.7.

I_CPN84CG_B17_5 : = 3
Available since version 1.7.

I_CPN84CG_B20 : = 4
Available since version 1.7.

I_CPN84CG_B25 : = 5
Available since version 1.7.

I_CPN84CG_B30 : = 6
Available since version 1.7.

I_CPN84CG_B35 : = 7
Available since version 1.7.

I_CPN84CG_B40 : = 8
Available since version 1.7.

I_CPN84CG_B50 : = 9
Available since version 1.7.

I_CPN84CG_B55 : = 10
Available since version 1.7.

I_CPN84CG_B60 : = 11
Available since version 1.7.

I_CPN84CG_B65 : = 12
Available since version 1.7.

IRConcr_PN84_ExposureRatings

Types of exposure rating defined for the Polish code PN-84.

Properties:

I_CPN84ER_MILD : = 0
Available since version 1.7.

I_CPN84ER_MODERATE : = 1
Available since version 1.7.

I_CPN84ER_SEVERE_OR_LEAKPROOF : = 2
Available since version 1.7.

IRConcr_PN99_ExposureRatings

Types of exposure rating defined for the Polish code PN-99.

Properties:

I_CPN99ER_1 : = 0
Available since version 1.7.

I_CPN99ER_2A : = 1
Available since version 1.7.

I_CPN99ER_2B : = 2

Available since version 1.7.

I_CPN99ER_3 : = 3
Available since version 1.7.

I_CPN99ER_4A : = 4
Available since version 1.7.

I_CPN99ER_4B : = 5
Available since version 1.7.

I_CPN99ER_5A : = 6
Available since version 1.7.

I_CPN99ER_5B : = 7
Available since version 1.7.

I_CPN99ER_5C : = 8
Available since version 1.7.

IRConcr_PN99_ReinforceData

Derived from IRConcrReinforceData
Reinforcement parameters for the Polish code PN-99.
Steel grades are described by the identifier set RConr_PN_SteeGrades, concrete classes by RConcr_PN99_ConcreteGrades, whereas types of exposure rating by RConcr_PN99_ExposureRatings.

Properties:

SLS_Cracking : bool
flag indicating if cracking width is to be calculated
Available since version 1.7.

SLS_Deflection : bool
flag indicating if a deflection value for RC slab taking cracking into account is to be calculated
Available since version 1.7.

SLS_CrackingReinfCorrect : bool
flag switching on automatic correction of cracking width by increasing area of reinforcement
Available since version 1.7.

SLS_DeflectionReinfCorrect : bool
flag switching on automatic correction of deflections by increasing reinforcement area
Available since version 1.7.

SLS_Exposure : short

exposure rating
Available since version 1.7.

SLS_MaxCracking : double
admissible cracking width
Available since version 1.7.

SLS_MaxDeflection : double
admissible deflection value
Available since version 1.7.

SLS_Concrete : double
concrete age (years)
Available since version 1.7.

SLS_ConcreteAge : short
concrete age expressed in days (it is significant to which range among the ranges determined by the code the day number belongs)
Available since version 1.7.

SLS_EnvHumidityVal : double
environment humidity value (in %)
Available since version 1.7.

Main : RConcrReinforceDataMain
SLS_LoadRatio : double
SLS_CreepingCoef : bool
SLS_CreepingCoefValue : double

IRConcr_PN99_ConcreteGrades

Concrete classes for Polish code PN-99

Properties:

I_CPN99CG_ANOTHER : = -1
other concrete class
Available since version 1.7.

I_CPN99CG_B15 : = 0
Available since version 1.7.

I_CPN99CG_B20 : = 1
Available since version 1.7.

I_CPN99CG_B25 : = 2
Available since version 1.7.

I_CPN99CG_B30 : = 3
Available since version 1.7.

I_CPN99CG_B37 : = 4
Available since version 1.7.

I_CPN99CG_B45 : = 5
Available since version 1.7.

I_CPN99CG_B50 : = 6
Available since version 1.7.

I_CPN99CG_B55 : = 7
Available since version 1.7.

I_CPN99CG_B60 : = 8
Available since version 1.7.

I_CPN99CG_B65 : = 9
Available since version 1.7.

I_CPN99CG_B70 : = 10
Available since version 1.7.

IRConcr_BAEL_ReinforceData

Derived from IRConcrReinforceData
Parameters of plate and shell reinforcement according to the BAEL code (91 and 99). Steel grades are determined by the identifiers from the set RConcr_BAEL_SteelGrades whereas concrete classes by RConcr_BAEL_ConcreteGrades.

Properties:

Cracking : RConcr_BAEL_CrackingType
Available since version 1.7.

CrackEnv : RConcr_BAEL_EnvironmentType
additional parameter resulting from the requirements of DTU 14-1 or Fascicule 74
Available since version 1.7.

CrackWaterLevel : RConcr_BAEL_WaterLevel
additional parameter resulting from the requirements of DTU 14-1
Available since version 1.7.

CrackAlpha : double
additional parameter resulting from the requirements of DTU 14-1 or Fascicule 74
Available since version 1.7.

SteelSymbol : string
Available since version 1.7.

SLS_Deflection : bool

Available since version 1.7.

SLS_DeflectionReinfCorrect : bool
Available since version 1.7.

SLS_MaxDeflection : double
Available since version 1.7.

CrackExtraParams : bool
flag indicating if additional parameters resulting from the requirements of DTU 14-1 (for limited cracking)
or Fascicule 74 (for not permissible cracking) are defined
Available since version 2.0.

Main : RConcrReinforceDataMain
CrackingTop : RConcr_BAEL_CrackingType

Available since version 8.

CrackingBot : RConcr_BAEL_CrackingType

Available since version 8.

CrackEnvTop : RConcr_BAEL_EnvironmentType

Available since version 8.

CrackEnvBot : RConcr_BAEL_EnvironmentType

Available since version 8.

CrackWaterLevelTop : RConcr_BAEL_WaterLevel

Available since version 8.

CrackWaterLevelBot : RConcr_BAEL_WaterLevel

Available since version 8.

CrackAlphaTop : double

Available since version 8.

CrackAlphaBot : double

Available since version 8.

CrackExtraParamsTop : bool

Available since version 8.

CrackExtraParamsBot : bool

Available since version 8.

IRConcr_BAEL_CrackingType

Properties:

I_CBAELCT_OTHER : = -1
Available since version 1.7.

I_CBAELCT_NOT_SEVERE : = 0
Available since version 1.7.

I_CBAELCT_SEVERE : = 1
Available since version 1.7.

I_CBAELCT_VERY_SEVERE : = 2
Available since version 1.7.

IRConcr_BAEL_EnvironmentType

Properties:

I_CBAELET_AGGRESSIVE_WATER : = 1
Available since version 1.7.

I_CBAELET_NORMAL_WATER : = 0
Available since version 1.7.

IRConcr_BAEL_WaterLevel

Properties:

I_CBAELWL_LOW : = 0
Available since version 1.7.

I_CBAELWL_HIGH : = 1
Available since version 1.7.

I_CBAELWL_EXCEPTIONAL : = 2
Available since version 1.7.

IRConcr_BAEL_ConcreteGrades

Properties:

I_CBAELCG_OTHER : = -1
Available since version 1.7.

I_CBAELCG_C12_15 : = 0
Available since version 1.7.

I_CBAELCG_C16_20 : = 1
Available since version 1.7.

I_CBAELCG_C20_25 : = 2
Available since version 1.7.

I_CBAELCG_C25_30 : = 3
Available since version 1.7.

I_CBAELCG_C30_37 : = 4
Available since version 1.7.

I_CBAELCG_C35_45 : = 5
Available since version 1.7.

I_CBAELCG_C40_50 : = 6
Available since version 1.7.

I_CBAELCG_C45_55 : = 7
Available since version 1.7.

I_CBAELCG_C50_60 : = 8
Available since version 1.7.

IRConcr_BAEL_SteelGrades

Properties:

I_CBAELSG_PLAIN : = 0
Available since version 1.7.

I_CBAELSG_DEFORMED : = 1
Available since version 1.7.

IRConcr_PN84_HumidityType

Types of humidity defined for the PN-84 code

Properties:

I_CPN84HT_40 : = 0
up to 40 percent
Available since version 2.0.

I_CPN84HT_40_75 : = 1
from 40 to 75 percent
Available since version 2.0.

I_CPN84HT_75 : = 2
over 75 percent
Available since version 2.0.

I_CPN84HT_WATER : = 3
water
Available since version 2.0.

IRConcr_ACI318_ReinforceData

Interface of reinforcement parameters for the codes: "ACI 318/99" and "ACI 318/99 metric".
Available since version 3.5.

Properties:

SLS_Cracking : bool
flag determining calculation of cracking width
Available since version 3.5.

SLS_Deflection : bool
flag determining calculation of a deflection value
Available since version 3.5.

SLS_DeflectionReinfCorrection : bool
flag managing automatic correction of deflections by increasing the area of reinforcement
Available since version 3.5.

SLS_MaxCracking : double
admissible value of cracking width
Available since version 3.5.

SLS_MaxDeflection : double
admissible value of deflection
Available since version 3.5.

SLS_LoadRatio : double
ratio of long-term loads to total loads

Available since version 3.5.

SLS_LoadActionPeriod : IRConcr_ACI318_LoadActionPeriodType
duration of variable long-term loads
Available since version 3.5.

SLS_Factor : bool
flag indicating if the concrete creep coefficient should be consiered in reinforcement definition
Available since version 3.5.

SLS_FactorValue : double
value of concrete creep coefficient
Available since version 3.5.

IsMetric : bool
flag informing about the code variant: true value for "ACI 318/99 metric", false value for "ACI 318/99".
Available since version 3.5.

Main : RConcrReinforceDataMain
SLS_CreepingCoef : bool
SLS_CreepingCoefValue : double

Operations:

SetBarDim (_type : IRConcrReinforceBarType, _val : IRConcr_ACI318_BarDim)
Function sets a diameter of the specified bar.
Available since version 3.5.

GetBarDim (_type : IRConcrReinforceBarType) : IRConcr_ACI318_BarDim
Function takes diameter of the specified bar.
Available since version 3.5.

SetMetricBarDim (_type : IRConncrReinforceBarType, _val : IRConncr_ACI318_MetricBarDim)
Equivalent of the SetBarDim function for the metric variant of the code.
Available since version 3.5.

GetMetricBarDim (_bar_type : IRConncrReinforceBarType) : IRConncr_ACI318_MetricBarDim
Equivalent of the GetBarDim function for the metric variant of the code.
Available since version 3.5.

IRConcr_ACI318_LoadActionPeriodType

Available duration times of variable long-term loads.
Available since version 3.5.

Properties:

I_ACI318_LAPT_3 : = 0

Available since version 3.5.

I_ACI318_LAPT_6 : = 1

Available since version 3.5.

I_ACI318_LAPT_12 : = 2

Available since version 3.5.

I_ACI318_LAPT_60 : = 3

Available since version 3.5.

IRConcr_ACI318_SteelGrades

Available since version 3.5.

Properties:

I_ACI318_ST_40 : = 0

Available since version 3.5.

I_ACI318_ST_50 : = 1

Available since version 3.5.

I_ACI318_ST_60 : = 2

Available since version 3.5.

I_ACI318_ST_75 : = 3

Available since version 3.5.

IRConcr_ACI318_ConcreteParams

Derived from IRConcrConcreteParams
Parameters describing concrete for the code ACI 318/99.
Available since version 3.5.

Properties:

AsInStructure : bool

flag indicating if reinforcement definition considers compression resistance of the material currently addded to the panel
Available since version 3.5.

IRConcr_ACI318_BarDim

Available bar diameters for the code ACI 318/99.
Available since version 3.5.

Properties:

I_ACI318_BR_3 : = 0

Available since version 3.5.

I_ACI318_BR_4 : = 1
I_ACI318_BR_5 : = 2

Available since version 3.5.

I_ACI318_BR_6 : = 3

Available since version 3.5.

I_ACI318_BR_7 : = 4

Available since version 3.5.

I_ACI318_BR_8 : = 5

Available since version 3.5.

I_ACI318_BR_9 : = 6

Available since version 3.5.

I_ACI318_BR_10 : = 7

Available since version 3.5.

I_ACI318_BR_11 : = 8

Available since version 3.5.

I_ACI318_BR_14 : = 9

Available since version 3.5.

I_ACI318_BR_18 : = 10

Available since version 3.5.

IRConcr_ACI318_MetricBarDIm

Available bar diameters for the code ACI 318/99 metric.
Available since version 3.5.

Properties:

I_ACI318_MBD_5 : = 0

Available since version 3.5.

I_ACI318_MBD_6 : = 1

Available since version 3.5.

I_ACI318_MBD_8 : = 2

Available since version 3.5.

I_ACI318_MBD_10 : = 3

Available since version 3.5.

I_ACI318_MBD_12 : = 4

Available since version 3.5.

I_ACI318_MBD_14 : = 5

Available since version 3.5.

I_ACI318_MBD_16 : = 6

Available since version 3.5.

I_ACI318_MBD_20 : = 7

Available since version 3.5.

I_ACI318_MBD_25 : = 8

Available since version 3.5.

I_ACI318_MBD_32 : = 9

Available since version 3.5.

I_ACI318_MBD_40 : = 10

Available since version 3.5.

IRConcrReinforceBarType

Types (designations) of reinforcing bars singled out during reinforcement definition.
Available since version 3.5.

Properties:

I_RBT_D1_TOP : = 0

Available since version 3.5.

I_RBT_D1_BOTTOM : = 1

Available since version 3.5.

I_RBT_D2_TOP : = 2

Available since version 3.5.

I_RBT_D2_BOTTOM : = 3

Available since version 3.5.

IRConcr_BS8110_ReinforceData

Interface of reinforcement parameters for the code BS8110.
Available since version 3.5.

Properties:

SLS_Cracking : bool
flag determinig calculation of cracking width
Available since version 3.5.

SLS_Deflection : bool
flag determinig calculation of a deflection value
Available since version 3.5.

SLS_DeflectionReinfCorection : bool
flag managing automatic correction of deflections by increasing reinforcement area
Available since version 3.5.

SLS_CrackingReinfCorection : bool
flag managing automatic correction of cracking width by increasing reinforcement area
Available since version 3.5.

SLS_Exposure : IRConcr_BS8110_ExposureRatings
exposure
Available since version 3.5.

SLS_MaxDeflection : double
admissible deflection value
Available since version 3.5.

SLS_MaxCracking : double
admissible value of cracking width
Available since version 3.5.

SLS_LoadRatio : double
ratio of long-term loads to total loads
Available since version 3.5.

SLS_ConcreteAge : IRConcr_BS8110_ConcreteAge
concrete age (days)
Available since version 3.5.

SLS_EnvHumidity : short
humidity (%)
Available since version 3.5.

SLS_CreepingCoef : bool
flag indicating if concrete creep coefficient should be considered in reinforcement definition
Available since version 3.5.

SLS_CreepingCoefValue : double
concrete creep coefficient
Available since version 3.5.

PartialSafetyFactors : IRConcr_BS8110_PartialSafetyFactors
safety factor
Available since version 3.5.

Main : RConcrReinforceDataMain

IRConcr_BS8110_ExposureRatings

Exposure types defined for the code BS 8110.
Available since version 3.5.

Properties:

I_BS8110_ER_MILD : = 0

Available since version 3.5.

I_BS8110_ER_MODERATE : = 1

Available since version 3.5.

I_BS8110_ER_SEVERE : = 2

Available since version 3.5.

I_BS8110_ER_VERY_SEVERE : = 3

Available since version 3.5.

I_BS8110_ER_MOST_SEVERE : = 4

Available since version 3.5.

IRConcr_BS8110_ConcreteAge

Predicted values of concrete age.
Available since version 3.5.

Properties:

I_BS8110_CA_1 : = 0
1 day
Available since version 3.5.

I_BS8110_CA_3 : = 1
3 days
Available since version 3.5.

I_BS8110_CA_7 : = 2
7 days
Available since version 3.5.

I_BS8110_CA_28 : = 3
28 days
Available since version 3.5.

I_BS8110_CA_90 : = 4
90 days
Available since version 3.5.

I_BS8110_CA_365 : = 5
365 days
Available since version 3.5.

IRConcr_BS8110_ConcreteGrades

Concrete classes according to the code BS 8110.
Available since version 3.5.

Properties:

I_BS8110_CG_AUTO : = 0

Available since version 3.5.

I_BS8110_CG_C12_15 : = 1

Available since version 3.5.

I_BS8110_CG_C16_20 : = 2

Available since version 3.5.

I_BS8110_CG_C20_25 : = 3

Available since version 3.5.

I_BS8110_CG_C25_30 : = 4

Available since version 3.5.

I_BS8110_CG_C30_37 : = 5

Available since version 3.5.

I_BS8110_CG_C35_45 : = 6

Available since version 3.5.

I_BS8110_CG_C40_50 : = 7

Available since version 3.5.

I_BS8110_CG_C45_55 : = 8

Available since version 3.5.

I_BS8110_CG_C50_60 : = 9

Available since version 3.5.

IRConcr_BS8110_PartialSafetyFactors

Safety factor types.

Available since version 3.5.

Properties:

I_BS8110_PSF_1985 : = 0

Available since version 3.5.

I_BS8110_PSF_1997 : = 1

Available since version 3.5.

IRConcrSlabRequiredReinfCalcParams

Calculation parameters for the required (theoretical) reinforcement for plates and shells.
Available since version 5.5.

Properties:

Method : IRobotReinforceCalcMethods
calculation method
Available since version 5.5.

GloballyAvgDesginForces : bool

Available since version 5.5.

ForcesReduction : bool

Available since version 5.5.

DisplayErrors : bool

Available since version 5.5.

Panels : IRobotSelection
selection of objects for which calculations will be performed
Available since version 5.5.

CasesULS : IRobotSelection

Available since version 5.5.

CasesSLS : IRobotSelection

Available since version 5.5.

CasesACC : IRobotSelection

Available since version 5.5.

IRConcrSlabRequiredReinfEngine

Calculation module for required (theoretical) reinforcement of plates and shells.
Available since version 5.5.

Properties:

Params : RConcrSlabRequiredReinfCalcParams
calculation parameters
Available since version 5.5.

Operations:

Calculate () : bool
Function performs calculations of the required (theoretical) reinforcement of plates and shells according to the currently-set parameters.
Available since version 5.5.

IRConcrReinforceCalcType

Properties:

I_CRCT_SIMPLE_BENDING : = 1
I_CRCT_BENDING_COMPRESSION_TENSION : = 2
I_CRCT_COMPRESSION_TENSION : = 3

IRConcrReinforceData2

Derived from IRConcrReinforceData

Properties:

Main : RConcrReinforceDataMain

IRConcrReinforceDataMain

Derived from IRConcrReinforceData

Properties:

ReinfCalcType : RConcrReinforceCalcType
UnidirReinf : bool
unidirectional reinforcement

MembraneReinfInOneLayer : bool
membrane reinforcement in one layer

IRConcr_SNIP_ReinforceData

Derived from IRConcrReinforceData2

Interface of plate/shell reinforcement parameters for the code "SNIP 2.03.01-84".

Properties:

SLS_Cracking : bool
SLS_CrackingReinfCorrection : bool
SLS_Exposure : RConcr_SNIP_Exposure

IRConcr_SNIP_SteelGrades

Properties:

I_CSNIPSG_A_I : = 3
I_CSNIPSG_A_II : = 4
I_CSNIPSG_A_III : = 5
I_CSNIPSG_A_IV : = 6
I_CSNIPSG_A_V : = 7
I_CSNIPSG_A_VI : = 8
I_CSNIPSG_A_IIIB_YH : = 9
I_CSNIPSG_A_IIIB_Y : = 10
I_CSNIPSG_BP_I : = 11
I_CSNIPSG_B_II : = 12
I_CSNIPSG_BP_II : = 13
I_CSNIPSG_K_7 : = 14
I_CSNIPSG_K_19 : = 15

IRConcr_SNIP_ConcreteGrades

Properties:

I_CSNIPCG_B7_5 : = 13
I_CSNIPCG_B10 : = 0
I_CSNIPCG_B12_5 : = 1
I_CSNIPCG_B15 : = 2
I_CSNIPCG_B20 : = 4
I_CSNIPCG_B25 : = 5
I_CSNIPCG_B30 : = 6
I_CSNIPCG_B35 : = 7
I_CSNIPCG_B40 : = 8
I_CSNIPCG_B45 : = 14
I_CSNIPCG_B50 : = 9
I_CSNIPCG_B55 : = 10
I_CSNIPCG_B60 : = 11

IRConcr_SNIP_ConcreteTypes

Properties:

I_CSNIPCT_HEAVYWEIGHT : = 1
I_CSNIPCT_FINE_GRAINED_A : = 2
I_CSNIPCT_FINE_GRAINED_B : = 3
I_CSNIPCT_FINE_GRAINED_V : = 4
I_CSNIPCT_LIGHTWEIGHT_SYNTHETIC_COMPACT : = 5
I_CSNIPCT_LIGHTWEIGHT_SYNTHETIC_COARSE_GRAINED : = 6
I_CSNIPCT_LIGHTWEIGHT_NATURAL_FILLER : = 7
I_CSNIPCT_POROUS : = 8
I_CSNIPCT_CELLULAR : = 9

IRConcr_SNIP_CuringMethods

Properties:

I_CSNIPCM_NORMAL : = 0
I_CSNIPCM_THERMAL_TREATMENT : = 1
I_CSNIPCM_AUTOCLAVES : = 2

IRConcr_SNIP_ConcreteParams

Derived from IRConcrConcreteParams

Properties:

Type : RConcr_SNIP_ConcreteTypes
CuringMethod : RConcr_SNIP_CuringMethods
ConcretingInLayers : bool
HighHumidity : bool

IRConcr_SNIP_Exposure

Properties:

I_CSNIPE_INTERNAL : = 0
I_CSNIPE_EXTERNAL : = 1
I_CSNIPE_GROUND_VARIABLE_WATER_LEVEL : = 2

IRConcr_EC2_ConcreteGrades

Concrete classes according to Eurocode 2.

Properties:

I_CEC2CG_AUTOMATIC : = -1
material defined for a panel will be used

I_CEC2CG_C12_15 : = 0
I_CEC2CG_C16_20 : = 1
I_CEC2CG_C20_25 : = 2
I_CEC2CG_C25_30 : = 3
I_CEC2CG_C30_37 : = 4
I_CEC2CG_C35_45 : = 5
I_CEC2CG_C40_50 : = 6
I_CEC2CG_C45_55 : = 7
I_CEC2CG_C50_60 : = 8

IRConcr_EC2_ITALIAN_SteelGrades

Properties:

I_CEC2ITSG_FE_B_22 : = 0
I_CEC2ITSG_FE_B_32 : = 1
I_CEC2ITSG_FE_B_38 : = 2
I_CEC2ITSG_FE_B_44 : = 3

IRConcr_EC2_ExposureRatings

Exposure types according to Eurocode 2.

Properties:

I_CEC2ER_1 : = 0
I_CEC2ER_2A : = 1
I_CEC2ER_2B : = 2
I_CEC2ER_3 : = 3
I_CEC2ER_4A : = 4
I_CEC2ER_4B : = 5
I_CEC2ER_5A : = 6
I_CEC2ER_5B : = 7
I_CEC2ER_5C : = 8

IRConcr_EC2_ReinforceData

Derived from IRConcrReinforceData2

Properties:

SLS_Cracking : bool
SLS_Deflection : bool
SLS_CrackingReinfCorrect : bool
SLS_DeflectionReinfCorrect : bool
SLS_Exposure : RConcr_EC2_ExposureRatings
SLS_MaxCracking : double
SLS_MaxDeflection : double
SLS_ConcreteAge : short

SLS_EnvHumidityVal : double
SLS_CreepingCoef : bool
SLS_CreepingCoefValue : double
NAD : RConcr_EC2_NAD

IRConcr_EC2_NAD

Properties:

I_CEC2NAD_BELGIAN : = 707
I_CEC2NAD_DUTCH : = 706
I_CEC2NAD_FINNISH : = 716
I_CEC2NAD_FRENCH : = 710
I_CEC2NAD_GERMAN : = 712
I_CEC2NAD_ITALIAN : = 714

Available since version 7.5.

IRConcrMemberRequiredReinfEngine

Calculation module for the design of required reinforcement of RC members.
Available since version 7.5.

Properties:

Params : RConcrMemberRequiredReinfCalcParams
Calculation parameters
Available since version 7.5.

Operations:

Calculate () : bool
Function performs calculations of the required reinforcement of RC members according to the currently-set parameters.
Available since version 7.5.

IRConcrMemberRequiredReinfCalcParams

Calculation parameters for the design of required reinforcement of RC members.
Available since version 7.5.

Properties:

Members : IRobotSelection
bar selection
Available since version 7.5.

CasesULS : IRobotSelection

Available since version 7.5.

CasesSLS : IRobotSelection

Available since version 7.5.

CasesALS : IRobotSelection

Available since version 7.5.

BeamPointsType : RConcrBeamCalcPointDefinitionType
method of definition of division points for a beam
Available since version 7.5.

BeamPointsValue : double
value determining points of the beam division
Available since version 7.5.

CombULS : bool

Available since version 7.5.

CombSLS : bool

Available since version 7.5.

CombALS : bool

Available since version 7.5.

IRConcrBeamCalcPointDefinitionType

Available methods of definition of calculation points for a beam.
Available since version 7.5.

Properties:

I_CBCPDT_POINT_COUNT : = 0
number of points for beam
Available since version 7.5.

I_CBCPDT_POINT_SPACING : = 1
distance between successive division points expressed in meters
Available since version 7.5.

Theoretical reinforcement for concrete bars

IRBestCodeService

RBestCodeService makes code-dependent definitions and calculations for theoretical reinforcement of RC members available.

Properties:

CalcParamsDlg : IRBestCalcParamsDlg
a dialog box that allows to edit calculation parameters

MemberDlg : IRBestMemberDlg
a dialog box that allows to edit RC member parameters

CalcEngine : IRBestCodeCalcEngine
the object performing code calculations

IRBestCalcParamsDlg

The interface describes the dialog box that allows the parameter definition for theoretical reinforcement calculations for RC members.

Operations:

DoModal (param_list : IRBestCalcParamsDataList)
The function creates and displays a new modal dialog box that allows to edit the list of objects describing calculation parameters.

SetStandard (params : IRBestCalcParamsData)
The function stets standard (default) parameter values for the object.

IRBestMemberDlg

The interface describes the dialog box that makes a code-dependent definition of a RC member type possible.

Operations:

DoModal (params : IRBestMemberData)
The function creates and displays a modal dialog box that allows to edit the bar code parameters saved in a given object.

SetStandard (params : IRBestMemberData)
The function sets standard (default) values in the indicated object for the code-dependent parameters.

IRBestParamSet

The interface describes a set of parameters that can be divided into two categories: values with a well defined meaning independent from code and these ones that meaning is connected with the given code. All values are available by the function set of the type Get/Set. The identifier of the taken or set quantity is their parameter. Code-dependent parameters have identifiers beginning from USER_VALUE upwards.

Operations:

GetDouble (param_id : int, value : double*) : bool

The function takes a real number, which is a parameter value with a given identifier. The function returns zero (False) if the parameter value with the indicated identifier was not set.

SetDouble (param_id : int, value : double) : bool

The function sets a value of the given parameter if it is correct (according to IsValidDouble). If the parameter value is incorrect, the function returns zero (False).

GetInteger (param_id : int, value : int*) : bool

The function takes an integer number, which is a parameter value with a given identifier. It gives back zero (False) if the value of the parameter with the indicated identifier was not set.

SetInteger (param_id : int, value : int) : bool

The function sets a value of the given parameter if it is correct (according to IsValidInteger). If the parameter value is incorrect, the function returns zero (False).

GetString (param_id : int, value : BSTR*) : bool

The function takes a text (character string) which is a parameter value with the given identifier. It gives zero (False) back if the value of the parameter with the indicated identifier was not set.

SetString (param_id : int, value : BSTR) : bool

The function sets the value of the given parameter, if correct (according to IsValidString). If the parameter value is not correct, the function returns zero (False).

Clear ()

The function deletes all settings of parameters carried out by means of the group of functions Set*.

ClearInteger (param_id : int)

The function deletes the setting of the integer type parameter with the given identifier.

ClearDouble (param_id : int)

The function deletes the setting of the real-number type parameter with the given identifier.

ClearString (param_id : int)
The function deletes the setting of the text type parameter with the given identifier.

IsValidDouble (param_id : int, value : double) : bool
The function checks if the indicated real number is a correct value of the parameter with the indicated identifier. It returns zero (False) if the value is not correct.

IsValidInteger (param_id : int, value : int) : bool
The function checks if the indicated integer is a correct value of the parameter with the indicated identifier. It returns zero (False) if the value is not correct.

IsValidString (param_id : int, value : string) : bool
The function checks if the indicated text (character string) is a correct value of the parameter with the indicated identifier. It returns zero (False) if the value is not correct.

IRBestCalcParamsData

Derived from IRBestParamSet
The interface defines calculation parameters of theoretical reinforcement for an RC member. Available parameters are described by the following types: RBestCalcParamsDataDoubleValue, RBestCalcParamsDataIntegerValue and RBestCalcParamsDataStringValue.

Properties:

ModularityList : IRobotValuesArray
a list of modularity values

IRBestCalcParamsDataDoubleValue

A set of parameter identifiers of real number type for the RBestCalcParamsData interface. The values of the set may be used as arguments defining the parameter type for the following functions of RBestCalcParamsData interface: SetDouble and GetDouble.

Properties:

I_BCPDDV_LONG_TENSION_BAR_DIAM : = 0
diameter of top reinforcing bars

I_BCPDDV_LONG_COMPRESSION_BAR_DIAM : = 1
diameter of bottom reinforcing bars

I_BCPDDV_TRAN_BAR_DIAM : = 2
diameter of transversal reinforcement

I_BCPDDV_TRAN_INCLINATION : = 3
a transversal reinforcement inclination angle (in radians)

I_BCPDDV_SAND_CONTENT : = 4
sand percentage [0..1] (for lightweight concrete)

I_BCPDDV_LONG_STEEL_FE : = 5
steel resistance of longitudinal reinforcement

I_BCPDDV_TRAN_STEEL_FE : = 6
steel resistance of transversal reinforcement

I_BCPDDV_CONCRETE_FC : = 7
concrete resistance

I_BCPDDV_COVER_SIZE : = 8
cover (calculated to bar axis)

I_BCPDDV_CONCRETE_CREEP : = 9
concrete creep coefficient

I_BCPDDV_REDISTRIBUTION : = 10
redistribution coefficient

I_BCPDDV_MAX_BREAK : = 11
maximum cracking width

I_BCPDDV_USER_VALUE : = 1000
the smallest identifier reserved for the user-defined values.

I_BCPDDV_MAX_DEFLECTION : = 12
Available since version 2.0.

I_BCPDDV_MAX_CRACK : = 13
Available since version 2.0.

I_BCPDDV_MAIN_RNF_COORD_X : = 14
Available since version 2.0.

I_BCPDDV_MAIN_RNF_COORD_Y : = 15
Available since version 2.0.

I_BCPDDV_MAIN_RNF_COORD_Z : = 16
Available since version 2.0.

I_BCPDDV_FLT2F : = 17
Available since version 2.0.

IRBestCalcParamsDataIntegerValue

An identifier set for integer type parameters for the RBestCalcPĺaramsData interface.

Properties:

I_BCPDIV_SEISMICS : = 0
seismic risk

I_BCPDIV_CRACKING : = 1
an exposure type

I_BCPDIV_TRAN_TYPE : = 2
type of transversal reinforcement (0 - stirrups, 1 - spiral)

I_BCPDIV_LEGS_NUM : = 3
number of legs in stirrups

I_BCPDIV_CONCRETE_LIGHT : = 4
lightweight concrete used (0 - no, 1 - yes)

I_BCPDIV_TRAN_ZONES_NUM : = 5
number of transversal reinforcement sections in final results (refers to beams)

I_BCPDIV_TRAN_ZONES_OPTIMALIZATION : = 6
optimization of spacing of transversal reinforcement sections (0 - switched off, 1 - switched on)

I_BCPDIV_LONG_STEEL_BY_CLASS : = 7
steel resistance of longitudinal reinforcement defined on the basis of steel grade (0 - no, 1 - yes)

I_BCPDIV_TRAN_STEEL_BY_CLASS : = 8
steel resistance of transversal reinforcement defined on the basis of steel grade (0 - no, 1 - yes)

I_BCPDIV_CONCRETE_BY_CLASS : = 9
concrete resistance defined on the basis of a concrete class (0 - no, 1 - yes)

I_BCPDIV_CONCRETE_BY_CONSTRUCTION : = 10
concrete resistance taken from the structure (0 - no, 1 - yes)

I_BCPDIV_LONG_BAR_DIAM_THE_SAME : = 11
equal bar diameters of a top and bottom longitudinal reinforcement (0 - no, 1 - yes)

I_BCPDIV_USER_VALUE : = 1000
the smallest identifier reserved for the user-defined values

I_BCPDIV_CALC_DEFLECTION : = 12
Available since version 2.0.

I_BCPDIV_CALC_CRACK : = 13
Available since version 2.0.

I_BCPDIV_ADJUST_DEFLECTION : = 14
Available since version 2.0.

I_BCPDIV_FLT2F : = 15
Available since version 2.0.

I_BCPDIV_MAIN_RNF_COORD_SYSTEM : = 16
Available since version 2.0.

IRBestCalcParamsDataStringValue

The identifier set for text values transferred for the RBestCalcParamsData interface.

Properties:

I_BCPDSV_NAME : = 0
name of parameter set

I_BCPDSV_LONG_REINF_CLASS : = 1
name of steel grade of longitudinal reinforcement

I_BCPDSV_TRAN_REINF_CLASS : = 2
name of steel grade of transversal reinforcement

I_BCPDSV_CONCRETE_CLASS : = 3
name of concrete class

I_BCPDSV_USER_VALUE : = 1000
the smallest identifier reserved for the user-defined values

IRBestCalcParamsDataList

An object list of the BestCalcParamsData type. Each list element is associated with a label name. All the names are saved in the LabNames table.

Properties:

Selected : string
name of the label associated with the indicated element from the list (if the name is empty, no element from the list is indicated)

LabNames : IRobotNamesArray
table with label names associated with particular sets of parameters

Operations:

Get (lab_name : string) : IRBestCalcParamsData
The function gives a list element associated with the given label

Delete (lab_name : string) : bool
The function deletes from the list an element associated with the given label

Store (lab_name : string, params : IRBestCalcParamsData) : bool
The function writes the defined set of parameters as the data for the label of the defined name. Definition of a new name results in the creation of a new label associated with the defined set of parameters. If an existing name is selected, the data associated with it will be overwritten.

Create () : IRBestCalcParamsData
The function creates and returns a new empty set of parameters.

IRBestMemberData

Derived from IRBestParamSet
The interface describes a data set for a type of RC member. Accessible parameters are described by the following types: RBestMemberDataDoubleValue, RBestMemberDataIntegerValue and RBestMemberDataStringValue.

IRBestMemberDataDoubleValue

A set of identifiers of real number parameters for the RBestMemberData interface.

Properties:

I_BMDDV_LENGTH_Y : = 0
length in Y direction

I_BMDDV_LENGTH_Z : = 1
length in Z direction

I_BMDDV_BUCK_COEF_Y : = 2
buckling coefficient Y

I_BMDDV_BUCK_COEF_Z : = 3
buckling coefficient Z

I_BMDDV_SUPPORT_WIDTH_1 : = 4
length of the first support (calculated to axis)

I_BMDDV_SUPPORT_WIDTH_2 : = 5
length of the second support (calculated to axis)

I_BMDDV_USER_VALUE : = 1000
the smallest identifier reserved for the user-defined values

IRBestMemberDataIntegerValue

An identifier set of integer type parameters for the BestMemberData interface.

Properties:

I_BMDIV_CALC_AS_SLENDER_Y : = 1
calculations of slender columns in direction Y (0 - no, 1 - yes)

I_BMDIV_SWAY_Y : = 3
sway structure in direction Y (0 - no, 1 - yes)

I_BMDIV_SWAY_Z : = 4
sway structure in direction Z (0 - no, 1 - yes)

I_BMDIV_USER_VALUE : = 1000
the smallest identifier reserved for a user-defined value

I_BMDIV_TYPE : = 0
assumes one of the values of the calculable type IRBestMemberType

I_BMDIV_CALC_AS_SLENDER_Z : = 2
calculations of slender columns in direction Z (0 - no, 1 - yes)

IRBestResults

Derived from IRBestParamSet
The interface describes calculation results of theoretical reinforcement for single section of RC member. Available values are described by the types: RBestResultsDoubleValue, RBestResultsIntegerValue and RBestResultsStringValue.

IRBestResultsDoubleValue

A set of identifiers of real-number parameters for the RBestResults interface.

Properties:

I_BRDV_AS_TOP : = 0
top reinforcement area

I_BRDV_AS_BOTTOM : = 1
bottom reinforcement area

I_BRDV_AS_LEFT : = 2
left reinforcement area

I_BRDV_AS_RIGHT : = 3
right reinforcement area

I_BRDV_STIRR_SPACE : = 9
stirrup spacing

I_BRDV_STIRR_SPACE_MOD : = 10
stirrup spacing with modularity taken into consideration

I_BRDV_LAMBDA_Y : = 13
lambda coefficient for Y direction

I_BRDV_LAMBDA_Z : = 14
lambda coefficient for Z direction

I_BRDV_CRACK_WIDTH : = 15
cracking width

I_BRDV_USER_VALUE : = 1000
the smallest identifier reserved for user-defined values.

I_BRDV_AS_MIN : = 4
the minimum code-defined reinforcement area

I_BRDV_AS_MAX : = 5
the maximum code-defined reinforcement area

I_BRDV_RO : = 6
reinforcement density

I_BRDV_RO_MIN : = 7
minimum code-defined reinforcement density

I_BRDV_RO_MAX : = 8
maximum code-defined reinforcement density

I_BRDV_STIRR_SPACE_MIN : = 11
minimum code-defined stirrup spacing

I_BRDV_STIRR_SPACE_MAX : = 12
maximum code-defined stirrup spacing

I_BRDV_RIGIDITY : = 16
Available since version 2.0.

I_BRDV_BAR_SPACING : = 17
Available since version 2.0.

IRBestResultsIntegerValue

A set of identifiers of real-number parameters for the interface RBestResults.

Properties:

I_BRIV_BEND_CASE_IDX : = 0
index of the designing load case in the force vector for the calculations carried out for bending

I_BRIV_BEND_ERROR_NUM : = 3
the error number (RBestCalcErrors determines available error numbers) for calculations for bending

I_BRIV_SHEAR_CASE_IDX : = 1
index of the designing load case in the force vector for the calculations carried out for shearing

I_BRIV_TORSION_CASE_IDX : = 2
index of the designing load case in the force vector for the calculations carried out for torsion

I_BRIV_USER_VALUE : = 1000
the smallest identifier reserved for user-defined values

I_BRIV_BEND_WARNING_NUM : = 6
warning bit mask for calculations carried out for bending (the bit values of the available warnings are determined by the interface RBestCalcWarnings)

I_BRIV_SHEAR_WARNING_NUM : = 7
warning bit mask for calculations carried out for shearing

I_BRIV_TORSION_WARNING_NUM : = 8
warning bit mask for calculations carried out for torsion

I_BRIV_SHEAR_ERROR_NUM : = 4
error number for calculations carried out for shearing

I_BRIV_TORSION_ERROR_NUM : = 5
error number for calculations carried out for torsion

IRBestCalcErrors

The set of identifiers describing errors in calculations of theoretical reinforcement.
If there occurred an error during calculations, the information about this fact should be written in the object of the RBestResults type, transferred as a parameter of the GetResults function of the RBestCodeCalcEngine interface. To write the information, one should use the SetInteger function that introduces integral values into an object. As an argument determining the type of integral value, one should provide the

relevant identifier of error type, while as the error value, one should provide the relevant value from the set RBestCalcErrors.

Properties:

I_BCE_NO_ERRORS : = 0
correct calculations

I_BCE_NO_CALC : = 1
no calculations

I_BCE_SECTION_TOO_SMALL : = 2
section values are too small for a defined moment

I_BCE_SECTION_SHEAR : = 3
section values are too small for a given shear force

I_BCE_BCKL_COLUMN : = 4
axial force reaches a critical value

I_BCE_USER_ERROR : = -1
an error (defined meaning)

IRBestCalcWarnings

Warning bit masks for calculations of theoretical reinforcement.

Properties:

I_BCW_NO_WARNINGS : = 0
no warnings

I_BCW_RO_MIN : = 0x1
a reinforcement ratio too low

I_BCW_RO_MAX : = 0x2
a reinforcement ratio too high

I_BCW_SPACE_BY_CODE : = 0x4
stirrup spacing calculated from a maximum stirrup spacing given in the code

I_BCW_LAMBDA_MAX : = 0x8
slenderness exceeds a maximum value given in the code

I_BCW_BREAK_MAX : = 0x10
cracking width exceeds a maximum value given in the code

I_BCW_USER_WARNING : = 0x20

warning (defined meaning)

IRBestResultsStringValue

Set of identifiers of text parameters for the interface RBestResults

Properties:

I_BRSV_BEND_ERROR_NAME : = 0
a message text about a definable error in calculations for bending

I_BRSV_BEND_WARNING_NAME : = 3
a message text about a definable warning in calculations for bending

I_BRSV_SHEAR_WARNING_NAME : = 4
a message text about a definable warning in calculations for shearing

I_BRSV_TORSION_WARNING_NAME : = 5
a message text about a definable warning in calculations for torsion

I_BRSV_SHEAR_ERROR_NAME : = 1
a message text about a definable error in calculations for shearing

I_BRSV_TORSION_ERROR_NAME : = 2
a message text about a definable error in calculations for torsion

I_BRSV_USER_VALUE : = 1000
the smallest identifier reserved for a user-defined value

IRBestForceData

Derived from IRBestParamSet
A set of forces acting in a bar section.

Properties:

Values : IRobotBarForceData
values of forces acting in a bar section.

LimitState : IRobotLimitState
type of analyzed limit state

IRBestDimParams

Derived from IRBestParamSet
A set of additional values parametrizing code (design) calculations. The RBestDimParamsIntegerValue type describes the set of available parameters.

IRBestDimParamsIntegerValue

Available identifiers of parameters of an integer type for the RBestDimParams interface.

Properties:

I_BDPIV_BEND_SIMPLE : = 0
calculations for simple bending (1 - on, 0 - off)

I_BDPIV_BEND_UNIAXIAL : = 1
calculations for offset compression and bending (1 - on, 0 - off)

I_BDPIV_BEND_BIAXIAL : = 2
calculations for offset biaxial compression (1 - on, 0 - off)

I_BDPIV_SHEAR : = 3
calculations for shearing (1 - on, 0 - off)

I_BDPIV_TORSION : = 4
calculations for torsion (1 - on, 0 - off)

I_BDPIV_USER_VALUE : = 1000
the smallest identifier reserved for the user-defined values

I_BDPIV_CALCULATION_TYPE : = 7
Available since version 2.0.

I_BDPIV_DIR_Y : = 5
Available since version 2.0.

I_BDPIV_DIR_Z : = 6
Available since version 2.0.

I_BDPIV_DIRECTION : = 8
Available since version 2.0.

I_BDPIV_BEND : = 9
Available since version 2.0.

I_BDPIV_LEVEL : = 10
Available since version 2.0.

IRBestCodeCalcEngine

The interface describes the object performing code calculations of theoretical reinforcement for RC members.

Operations:

SetParams (calc_params : IRBestCalcParamsData, memb_def : IRBestMemberData)
The function parametrizes the calculation process. It transfers information concerning parameters for calculation of theoretical reinforcement (calc_params) as well as RC member parameters (memb_def).

SetGeometry (sect_geo : RConcrBarSectionData)
The function parametrizes the calculation process. It provides information about the geometry of a bar section.

SetForces (forces : IRobotObjectsArray, pos : double)
The function parametrizes the calculation process in the case of short beams and columns. It provides information about forces acting in a section (forces), as well as the position of the section along the member (pos [0 .. member length]).

SetForcesSlender (forces_begin : IRobotObjectsArray, forces_midle : IRobotObjectsArray, forces_end : IRobotObjectsArray)
The function parametrizes the calculation process in the case of slender columns. It provides information about forces acting in column cross-sections: at column ends (forces_begin, forces_end) and at the intermediate cross-section (forces_middle)

Calculate (dim_params : IRBestDimParams)
The function performs calculations on a base of set parameters.

GetResults (results : IRBestResults)
The function returns the results of recently performed calculations to the indicated object.

IRBestMemberDataStringValue

The set of identifiers of text parameters for the RBestMemberData interface.

Properties:

I_BMDSV_USER_VALUE : = 1000
the smallest identifier reserved for the user

IRBestMemberType

A set of identifiers for a type of RC member or plate.

Properties:

I_BMT_BEAM : = 0
I_BMT_COLUMN : = 1
I_BMT_PLATE : = 2

IRBestPlateCalcParamsDlg

Operations:

DoModal (params : IRBestCalcParamsData, label_name : string)
Available since version 1.7.

SetStandard (params : IRBestCalcParamsData)
Available since version 1.7.

IRBestCodeCalculationType

Properties:

I_BCCT_PLATE : = 0
Available since version 1.7.

I_BCCT_MEMBER : = 1
Available since version 1.7.

IRBestCodeServiceExt

Derived from IRBestCodeService
Available since version 2.0.

Operations:

IsServed (calc_type : IRBestCodeCalculationType) : bool
Available since version 2.0.

GetCalcParamsDlg (calc_type : IRBestCodeCalculationType) : IUnknown
Function returns the interface to the appropriate dialog box for calculation parameter definition.
Available since version 2.0.

IRBestCalculationType

Properties:

I_BCT_DIMENSIONING : = 0
I_BCT_BAR_SPACING : = 1
I_BCT_RIGIDITY : = 2
I_BCT_CRACK_WIDTH : = 3

IRBestDimParamsDoubleValue

Properties:

I_BDPDV_AS : = 0
I_BDPDV_AS_TOP : = 1
I_BDPDV_AS_BOTTOM : = 2
I_BDPDV_AS_LEFT : = 3
I_BDPDV_AS_RIGHT : = 4
I_BDPDV_DIR_X : = 5
I_BDPDV_DIR_Y : = 6
I_BDPDV_DIR_Z : = 7
I_BDPDV_DIR_ALPHA : = 8
I_BDPDV_DIR_THETA : = 9
I_BDPDV_DIR_R : = 10
I_BDPDV_CONCRETE_FC : = 11
Available since version 2.0.

IRBestBendType

Properties:

I_BBT_FALSE : = 0
I_BBT_SIMPLE : = 1
I_BBT_UNIAXIAL : = 2
I_BBT_BIAXIAL : = 3

IRBestDirection

Properties:

I_BD_X : = 0
I_BD_Y : = 1
I_BD_Z : = 2

IRBestLevel

Properties:

I_BL_UP : = 0
I_BL_DOWN : = 1
Available since version 2.0.

IRBestForceDataDoubleValue

Properties:

I_BFDDV_COEF_LONG : = 0
Available since version 2.0.

IRBestCoordSystem

Properties:

I_BCS_CARTESIAN : = 1
Available since version 2.0.

I_BCS_POLAR : = 2
Available since version 2.0.

Steel and timber dimensioning

IRDimStreamType

Properties:

I_DST_LONG : = 1
I_DST_DOUBLE : = 2
I_DST_TEXT : = 3

IRDimStream

Operations:

Clear ()

Size (_type : IRDimStreamType) : long

SeekSet (_type : IRDimStreamType, pos : long)

WriteLong (val : long)

ReadLong () : long

WriteDouble (val : double)

ReadDouble () : double

WriteText (val : string)

ReadText () : string

IRDimMembDefType

Properties:

I_DMDT_USER : = 1
I_DMDT_MEMBER : = 2
I_DMDT_BEAM : = 3
I_DMDT_COLUMN : = 4

IRDimMembDefMatType

Properties:

I_DMDMT_STEEL : = 1
I_DMDMT_TIMBER : = 2
I_DMDMT_CONCRETE : = 3

Available since version 7.5.

IRDimMembDefLengthDataType

Properties:

I_DMDLDT_LENGTH_Y : = 1
I_DMDLDT_LENGTH_Z : = 2
I_DMDLDT_LENGTH_U : = 3
I_DMDLDT_LENGTH_V : = 4

IRDimMembDefBucklingDataType

Properties:

I_DMDBDT_BUCKLING_Y : = 1
I_DMDBDT_BUCKLING_Z : = 2
I_DMDBDT_BUCKLING_U : = 3
I_DMDBDT_BUCKLING_V : = 4

IRDimMembDefDispDataType

Properties:

I_DMDDDT_DISP_X : = 1
I_DMDDDT_DISP_Y : = 2

IRDimMembDefDeflDataType

Properties:

I_DMDDDT_DEFL_Y : = 3
I_DMDDDT_DEFL_Z : = 4

IRDimMembDef

Properties:

ClientID : string
MatType : IRDimMembDefMatType
Type : IRDimMembDefType
Length : double
Name : string

Operations:

Store (str : IRDimStream)

Retrieve (str : IRDimStream)

LengthYZUV (_type : IRDimMembDefLengthDataType) : double

IsBuckCoefConst (_type : IRDimMembDefBucklingDataType) : bool

IsDisplacementXY (_type : IRDimMembDefDispDataType) : bool

DisplXYRelLimit (_type : IRDimMembDefDispDataType) : double

IsDeflectionYZ (_type : IRDimMembDefDeflDataType) : bool

DeflYZRelLimit (_type : IRDimMembDefDeflDataType) : double

IRDimEffDefParamType

Properties:

I_DEDPT_ACC : = 1
I_DEDPT_ELEM_NO : = 2
I_DEDPT_CASE_NO : = 3
I_DEDPT_COMP_NO : = 4
I_DEDPT_POINTS_NUM : = 5
I_DEDPT_POINT_NO : = 6
I_DEDPT_LOADCLASS : = 7

IRDimEffDefDirType

Properties:

I_DEDDT_Y : = 1
I_DEDDT_Z : = 2

IRDimEffDefIntPsType

Properties:

I_DEDIPT_BUCKLING_Y : = 1
I_DEDIPT_BUCKLING_Z : = 2
I_DEDIPT_LBUCKLING_U : = 3
I_DEDIPT_LBUCKLING_L : = 4

IRDimEffDef

Properties:

N : double
QY : double
QZ : double
MX : double
MY : double
MZ : double

Operations:

Clear ()

WriteParam (_type : IRDimEffDefParamType, val : long)

ReadParam (_type : IRDimEffDefParamType) : long

WriteForces (N : double, QY : double, QZ : double, MX : double, MY : double, MZ : double)

WriteValuesSet1 (_type : IRDimEffDefDirType, M1 : double, M2 : double, M12 : double)

Read_M1 (_type : IRDimEffDefDirType) : double

Read_M2 (_type : IRDimEffDefDirType) : double

Read_M12 (_type : IRDimEffDefDirType) : double

WriteValuesSet2 (_type : IRDimEffDefDirType, MP_MAX : double, MN_MAX : double, M_MID : double, M_1P4L : double, M_3P4L : double)

Read_MP_MAX (_type : IRDimEffDefDirType) : double

Read_MN_MAX (_type : IRDimEffDefDirType) : double

Read_M_MID (_type : IRDimEffDefDirType) : double

Read_M_1P4L (_type : IRDimEffDefDirType) : double

Read_M_3P4L (_type : IRDimEffDefDirType) : double

WriteIntPsEffSet1 (_type : IRDimEffDefIntPsType, M1 : double, M2 : double, M12 : double)

Read_IntPsEff_M1 (_type : IRDimEffDefIntPsType) : double

Read_IntPsEff_M2 (_type : IRDimEffDefIntPsType) : double

Read_IntPsEff_M12 (_type : IRDimEffDefIntPsType) : double

WriteIntPsEffSet2 (_type : IRDimEffDefIntPsType, MP_MAX : double, MN_MAX : double, M_MID : double, M_1P4L : double, M_3P4L : double)

Read_IntPsEff_MP_MAX (_type : IRDimEffDefIntPsType) : double

Read_IntPsEff_MN_MAX (_type : IRDimEffDefIntPsType) : double

Read_IntPsEff_M_MID (_type : IRDimEffDefIntPsType) : double

Read_IntPsEff_M_1P4L (_type : IRDimEffDefIntPsType) : double

Read_IntPsEff_M_3P4L (_type : IRDimEffDefIntPsType) : double

IRDimMatDefType

Properties:

I_DMDT_NONE : = 0
I_DMDT_STEEL : = 1
I_DMDT_ALUMINIUM : = 2
I_DMDT_WOOD : = 3

IRDimMatDefValType

Properties:

I_DMDVT_CS : = 1
I_DMDVT_E : = 2
I_DMDVT_G : = 3
I_DMDVT_RE : = 4
I_DMDVT_RE_AX_COMR : = 5
I_DMDVT_NU : = 6
I_DMDVT_FU : = 7
I_DMDVT_LX : = 8

I_DMDVT_RO : = 9
I_DMDVT_RT : = 10
I_DMDVT_E_5 : = 11
I_DMDVT_TRANS : = 12
I_DMDVT_PN_E_TRANS : = 13
I_DMDVT_PN_E_ADITIONAL : = 14
I_DMDVT_RE_BENDING : = 15
I_DMDVT_RE_AX_TENS : = 16
I_DMDVT_RE_TR_TENS : = 17
I_DMDVT_RE_TR_COMPR : = 18
I_DMDVT_RE_SHEAR : = 19
I_DMDVT_DAMPCOEF : = 20

IRDimMatDefLongExValType

Properties:

I_DMDLEVT_TIMB_TYPE : = 1
I_DMDLEVT_CATEGORY : = 2
I_DMDLEVT_NATURE : = 3

IRDimMatDefDblExValType

Properties:

I_DMDDEVT_RETRAIT : = 4
I_DMDDEVT_HUMIDITY : = 5

IRDimMatDef

Properties:

Type : IRDimMatDefType
Name : string
SecondName : string

Operations:

SetNames (_name : string, second_name : string)

WriteValue (_type : IRDimMatDefValType, val : double)

ReadValue (_type : IRDimMatDefValType) : double

WriteLongExtraValue (_type : IRDimMatDefLongExValType, val : long)

ReadLongExtraValue (_type : IRDimMatDefLongExValType) : long

WriteDoubleExtraValue (_type : IRDimMatDefDblExValType, val : double)

ReadDoubleExtraValue (_type : IRDimMatDefDblExValType) : double

IRDimProfDefType

Properties:

I_DPDT_NONE : = 0
I_DPDT_CAE : = 1
I_DPDT_CAEP : = 2
I_DPDT_CAI : = 3
I_DPDT_CAIP : = 4
I_DPDT_DCEC : = 5
I_DPDT_DCED : = 6
I_DPDT_DCEP : = 7
I_DPDT_DCIG : = 8
I_DPDT_DCIP : = 9
I_DPDT_HEA : = 10
I_DPDT_HEAA : = 11
I_DPDT_HEB : = 12
I_DPDT_HEC : = 13
I_DPDT_HEM : = 14
I_DPDT_HER : = 15
I_DPDT_HHEA : = 16
I_DPDT_HHEB : = 17
I_DPDT_HHEM : = 18
I_DPDT_IIPE : = 19
I_DPDT_IPE : = 20
I_DPDT_IPEA : = 21
I_DPDT_IPEO : = 22
I_DPDT_IPER : = 23
I_DPDT_IPEV : = 24
I_DPDT_IPN : = 25
I_DPDT_MHEA : = 26
I_DPDT_MHEB : = 27
I_DPDT_MHEM : = 28
I_DPDT_MIPE : = 29
I_DPDT_PRS : = 30
I_DPDT_TCAR : = 31
I_DPDT_TEAE : = 32
I_DPDT_TEAI : = 33
I_DPDT_THEX : = 34
I_DPDT_TREC : = 35
I_DPDT_TRON : = 36
I_DPDT_UAP : = 37
I_DPDT_UPN : = 38
I_DPDT_UUAP : = 39
I_DPDT_UUPN : = 40
I_DPDT_OSIE : = 41
I_DPDT_CAISSON : = 42

I_DPDT_RECT : = 43
I_DPDT_DRECT : = 44
I_DPDT_TUBE : = 45
I_DPDT_ISYM : = 46
I_DPDT_INSYM : = 47
I_DPDT_TUSER : = 48
I_DPDT_CUSER : = 49
I_DPDT_FRTG : = 50
I_DPDT_CROSS : = 51
I_DPDT_X : = 52

IRDimProfDefItemType

Properties:

I_DPDIT_STANDARD : = 1
I_DPDIT_VAR_IN_POINT : = 2
I_DPDIT_VAR_MIDDLE : = 3
I_DPDIT_VAR_BEGEND : = 4

IRDimProfDefValType

Properties:

I_DPDVT_H : = 1
I_DPDVT_HW : = 2
I_DPDVT_HD : = 3
I_DPDVT_B : = 4
I_DPDVT_B2 : = 5
I_DPDVT_BF : = 6
I_DPDVT_BF2 : = 7
I_DPDVT_BD : = 8
I_DPDVT_EA : = 9
I_DPDVT_ES : = 10
I_DPDVT_EA2 : = 11
I_DPDVT_ES2 : = 12
I_DPDVT_DIS : = 13
I_DPDVT_R : = 14
I_DPDVT_R2 : = 15
I_DPDVT_VZ : = 16
I_DPDVT_VPZ : = 17
I_DPDVT_VY : = 18
I_DPDVT_VPY : = 19
I_DPDVT_I : = 20
I_DPDVT_IY : = 21
I_DPDVT_IZ : = 22
I_DPDVT_S : = 23
I_DPDVT_SY : = 24
I_DPDVT_SZ : = 25

I_DPDVT_MSY : = 26
I_DPDVT_MSZ : = 27
I_DPDVT_MASSE : = 28

IRDimProfDef

Properties:

Type : IRDimProfDefType
Name : string
IsVar : bool

Operations:

Clear ()

WriteValue (item : IRDimProfDefItemType, _type : IRDimProfDefValType, val : double)

ReadValue (item : IRDimProfDefItemType, _type : IRDimProfDefValType) : double

IRDimCalcStateFlagType

Properties:

I_DCSFT_CHECKSLEND : = 1
I_DCSFT_POINTMIDDLE : = 2
I_DCSFT_FIRE : = 3

IRDimCalcStateParamType

Properties:

I_DCSPT_CALCTYPE : = 1
I_DCSPT_CALCMETHOD : = 2

IRDimCalcStateParamValue

Properties:

I_DCSPV_MANUAL_VERIF : = 1
I_DCSPV_MEMBERS_VERIF : = 2
I_DCSPV_GROUP_VERIF : = 3
I_DCSPV_DIMENSIONING : = 4
I_DCSPV_OPTIMIZATION : = 5
I_DCSPV_CALCMETHOD_1 : = 100
I_DCSPV_CALCMETHOD_2 : = 101
I_DCSPV_CALCMETHOD_3 : = 102

IRDimCalcStateValueType

Properties:

I_DCSPV_MAXSLEND : = 1
I_DCSPV_EFFRATIO : = 2

IRDimCalcState

Operations:

SetFlag (_type : IRDimCalcStateFlagType, val : bool)

IsFlagSet (_type : IRDimCalcStateFlagType) : bool

SetParam (_type : IRDimCalcStateParamType, val : IRDimCalcStateParamValue)

GetParam (_type : IRDimCalcStateParamType) : IRDimCalcStateParamValue

SetParamValue (_type : IRDimCalcStateValueType, val : double)

GetParamValue (_type : IRDimCalcStateValueType) : double

IRDimMembSrv

Operations:

CheckLabelName (_name : string, is_defined : bool*, can_be_saved : bool*)

Save (val : IRDimMembDef)

IRDimMembCalcRetValue

Properties:

I_DMCRV_INCORRECTDATA : = -2
I_DMCRV_INSTABILITY : = -3
I_DMCRV_INCORRECT : = 0
I_DMCRV_CORRECT : = 1

IRDimMembCalcBuckType

Properties:

I_DMCBC_Y : = 1

I_DMCBC_Z : = 2

IRDimMembResTableLineType

Properties:

I_DMRTLT_PRINTCAPTION : = 0
I_DMRTLT_CAPTION : = 1
I_DMRTLT_SUBCAPTION : = 2
I_DMRTLT_1PARAM : = 3
I_DMRTLT_4HEADER : = 4
I_DMRTLT_5HEADER : = 5
I_DMRTLT_4PARAM : = 6
I_DMRTLT_5PARAM : = 7

IRDimMembResTableComp

Properties:

I_DMRTC_NAME : = 1
I_DMRTC_VALUE : = 2
I_DMRTC_UNIT : = 3
I_DMRTC_DESCRIPTION : = 4
I_DMRTC_PARAGRAPH : = 5

IRDimMembRes

Properties:

Language : long
ResOfCalc : IRDimMembCalcRetValue
SlendY : double
SlendZ : double
Ratio : double
EffRatio : double
RtfFileName : string
Units : IRDimUnits

Operations:

Retrieve (str : IRDimStream)

SetMembDef (memb_def : IRDimMembDef)

SetMatDef (mat_def : IRDimMatDef)

SetProfDef (prof_def : IRDimProfDef)

SetEffDef (eff_def : IRDimEffDef)

GetMembDefAccess () : IRDimMembDef

GetMatDefAccess () : IRDimMatDef

GetProfDefAccess () : IRDimProfDef

GetEffDefAccess () : IRDimEffDef

CreateResWnd (parent_hwnd : long) : long

GetMaxLineNo () : long

IsLineActive (line_no : long) : bool

GetLineType (line_no : long) : IRDimMembResTableLineType

GetLineComponent (line_no : long, cmpnt_no : IRDimMembResTableComp) : string

BlockCount (_name : string) : long

IsStatement (_name : string) : bool

ReplaceMark (mark : string, mark_out : string*) : bool

RecognizedPQ (_name : string, vstr : string*, ustr : string*) : bool

RefreshUnits (are_robot_units : bool)

IRDimMembCalc

Properties:

IsExtraMomentsSet1 : bool
IsExtraMomentsSet2 : bool
IsIntPointsMomentsSet1 : bool
IsIntPointsMomentsSet2 : bool

Operations:

SetMembDef (memb_def : IRDimMembDef)

SetMatDef (mat_def : IRDimMatDef)

SetProfDef (prof_def : IRDimProfDef)

SetEffDef (eff_data : IRDimEffDef)

SetCalcState (st : IRDimCalcState)

CalculBuckling ()

CalculMember () : IRDimMembCalcRetValue

GetRatio () : double

GetResultsInterface () : IRDimMembRes

IRDimCodeService

Operations:

GetDefaultMembDef (_type : IRDimMembDefType) : IRDimMembDef

EditMembDef (language : long, inModalWindow : bool, parentHWND : long, val : IRDimMembDef)

GetMembCalc () : IRDimMembCalc

IRDimClient

Operations:

GetRDimCodeService () : IRDimCodeService

IRDimUnitType

Properties:

I_DUT_SECDIMEN : = 1
I_DUT_SECSUR : = 2
I_DUT_SECVOL : = 3
I_DUT_SECMI : = 4
I_DUT_LENGTH : = 5
I_DUT_FORCE : = 6
I_DUT_DISPL : = 7
I_DUT_MOMENT : = 8
I_DUT_STRESS : = 9
I_DUT_NONE : = 10

IRDimUnits

Properties:

AreRobotUnits : bool

Operations:

Refresh (are_robot_units : bool)

ReadToUserCoef (unit_type : IRDimUnitType) : double

ReadUserName (unit_type : IRDimUnitType) : string

Format (unit_type : IRDimUnitType, val : double) : string

IRDimServer

Server responsible for design and verification of steel and timber members.
Available since version 2.5.

Code EC3

Available since version 3.5.

IRDimMembParamsEC3

Available since version 3.5.

Properties:

BucklingDiagramY : IRDimBuckDiagramEC3

Available since version 3.

BucklingDiagramZ : IRDimBuckDiagramEC3

Available since version 3.

LoadTypeY : IRDimLoadTypeEC3

Available since version 3.

LoadTypeZ : IRDimLoadTypeEC3

Available since version 3.

LoadLevel : IRDimLoadLevelEC3

Available since version 3.

LoadLevelValue : double

Available since version 3.

LatCoeffLowerFlange : IRDimLatBuckCoeffDiagramEC3

Available since version 3.

LatCoeffUpperFlange : IRDimLatBuckCoeffDiagramEC3

Available since version 3.

LatCoeffLowerFlangeValue : double

Available since version 3.

LatCoeffUpperFlangeValue : double

Available since version 3.

LatBuckType : IRDimLaterBuckTypeEC3

Available since version 3.

TensAreaNetGros : double

Available since version 3.

YieldStrengthValue : double

Available since version 3.

MaterCoeffGamma0 : double

Available since version 3.

MaterCoeffGamma1 : double

Available since version 3.

TubeControl : bool

Available since version 3.

BuckLengthCoeffY : double

Available since version 3.

BuckLengthCoeffZ : double

Available since version 3.

RelLimitDeflUy : double

Available since version 3.

RelLimitdeflUz : double

Available since version 3.

YieldStrengthType : IRDimYieldStrengthTypeEC3

Available since version 3.5.

HotRolledPipes : bool

Available since version 3.5.

IRDimLoadTypeEC3

Available since version 3.5.

Properties:

I_DLT_EC3_UNIFORM_LOAD : = 0

Available since version 3.

I_DLT_EC3_UNIFORM_MOMENT : = 1

Available since version 3.

I_DLT_EC3_CONCENTRATED_FORCE : = 2

Available since version 3.

I_DLT_EC3_MOMENTS_AT_ENDS : = 3

Available since version 3.

IRDimLaterBuckTypeEC3

Available since version 3.5.

Properties:

I_DLBT_EC3_SYMMETR_LOADED : = 0

Available since version 3.

I_DLBT_EC3_CANTILEVER : = 1

Available since version 3.

I_DLBT_EC3_NO : = 2

Available since version 3.

IRDimLoadLevelEC3

Available since version 3.5.

Properties:

I_DLL_EC3_UPP_EDGE_LOADED : = 0

Available since version 3.

I_DLL_EC3_UPP_SECT_PAR_LOADED : = 1

Available since version 3.

I_DLL_EC3_CENTER_LOADED : = 2

Available since version 3.

I_DLL_EC3_LOW_SEC_PART_LOADED : = 3

Available since version 3.

I_DLL_EC3_LOW_EDGE_LOADED : = 4

Available since version 3.

IRDimLatBuckCoeffDiagramEC3

Available since version 3.5.

Properties:

I_DLBCD_EC3_CANTILEVER_2_0 : = 0

Available since version 3.

I_DLBCD_EC3_INTERNAL_BRACINGS : = 5

Available since version 3.

I_DLBCD_EC3_NO : = 4

Available since version 3.

I_DLBCD_EC3_PINNED_PINNED_1_0 : = 1

Available since version 3.

I_DLBCD_EC3_STIFF_STIFF_0_5 : = 2

Available since version 3.

I_DLBCD_EC3_USER_DEFINED : = 3

Available since version 3.

IRDimBuckDiagramEC3

Available since version 3.5.

Properties:

I_DBD_EC3_AUTO : = 12

Available since version 3.

I_DBD_EC3_CANTILEVER_2_0 : = 3

Available since version 3.

I_DBD_EC3_INTERN_ADJBAR_6 : = 8

Available since version 3.

I_DBD_EC3_INTERNAL_BRACINGS : = 13

Available since version 3.

I_DBD_EC3_NO : = 11

Available since version 3.

I_DBD_EC3_PINNED_ADJBAR_1 : = 4

Available since version 3.

I_DBD_EC3_PINNED_ADJBAR_3 : = 6

Available since version 3.

I_DBD_EC3_PINNED_PINNED_1_0 : = 0

Available since version 3.

I_DBD_EC3_PINNED_STIFF_0_7 : = 2

Available since version 3.

I_DBD_EC3_STIFF_ADJBAR_1 : = 5

Available since version 3.

I_DBD_EC3_STIFF_ADJBAR_3 : = 7

Available since version 3.

I_DBD_EC3_STIFF_STIFF_0_5 : = 1

Available since version 3.

I_DBD_EC3_TRUSS_CHORD_0_9 : = 9

Available since version 3.

I_DBD_EC3_TRUSS_DIAGONAL_0_8 : = 10

Available since version 3.

I_DBD_EC3_USER_DEFINED : = -1

Available since version 3.

IRDimCodeResEC3

Available since version 3.5.

Properties:

IsBuckY : long

Available since version 3.

IsBuckZ : long

Available since version 3.

StrsShearY : double

Available since version 3.

StrsShearZ : double

Available since version 3.

StrsComp : double

Available since version 3.

StrsTens : double

Available since version 3.

StrsUprEdgeMY : double

Available since version 3.

StrsRgtEdgeMZ : double

Available since version 3.

StrsLowEdgeMY : double

Available since version 3.

StrsLftEdgeMZ : double

Available since version 3.

ClassOfSect : long

Available since version 3.

ClassOfSectElem1 : long

Available since version 3.

ClassOfSectElem2 : long

Available since version 3.

ClassOfSectElem3 : long

Available since version 3.

ClassOfSectElem4 : long

Available since version 3.

WebSlend : double

Available since version 3.

WebSlend1 : double

Available since version 3.

UppFlanSlend : double

Available since version 3.

LowFlanSlend1 : double

Available since version 3.

LowFlanSlend2 : double

Available since version 3.

BuckSLendLamy : double

Available since version 3.

BuclSlendLamz : double

Available since version 3.

BuckRelSlendLaby : double

Available since version 3.

BuckRelSlendLabz : double

Available since version 3.

LatBuckSlendLamlt : double

Available since version 3.

MaxBuckSlend : double

Available since version 3.

BuckParamKy : double

Available since version 3.

BuckParamKz : double

Available since version 3.

LatBuckParamKlt : double

Available since version 3.

PlastAxForcStrenNplrd : double

Available since version 3.

TensStrenNurd : double

Available since version 3.

PlastTensStrenNtrd : double

Available since version 3.

PlastCompStrenNcrd : double

Available since version 3.

BuckStrenNbyrd : double

Available since version 3.

BuckStrenNbzrd : double

Available since version 3.

OverallBuckStrenNbrd : double

Available since version 3.

LatBuckLengthLd : double

Available since version 3.

LatBuckParamLd : long

Available since version 3.

InteractParamAlfa : double

Available since version 3.

InteractParamBeta : double

Available since version 3.

LatBuckParamC1 : double

Available since version 3.

LatBuckParamC2 : double

Available since version 3.

WarpingConstantIw : double

Available since version 3.

TorsMomInertIt : double

Available since version 3.

InteractParamMiy : double

Available since version 3.

InteractParamMiz : double

Available since version 3.

InteractParamMilt : double

Available since version 3.

PlastMomStrenMplyrd : double

Available since version 3.

PlastMomStrenMplzrd : double

Available since version 3.

ElastMomStrenMelyrd : double

Available since version 3.

ElastMomStrenMelzrd : double

Available since version 3.

ReducMomStrenMcyrd : double

Available since version 3.

ReducMomStrenMczrd : double

Available since version 3.

ReducMomStrenMnyrd : double

Available since version 3.

ReducMomStrenMnzrd : double

Available since version 3.

ReducMomStrenMvyrd : double

Available since version 3.

ReducMomStrenMvzrd : double

Available since version 3.

LatBuckMomStrenmbrd : double

Available since version 3.

EffectiveMomenMeff : double

Available since version 3.

CritMomenMcr : double

Available since version 3.

ShearStrenTplyrd : double

Available since version 3.

ShearStrenTplzrd : double

Available since version 3.

BuckCoeffXy : double

Available since version 3.

BuckCoeffXz : double

Available since version 3.

BuckCoeffMinXi : double

Available since version 3.

LatBuckParamFilt : double

Available since version 3.

LatBuckCoeffXlt : double

Available since version 3.

LaodLevel : double

Available since version 3.

MaterCoeffGamma0 : double

Available since version 3.

MaterCoeffGamma1 : double

Available since version 3.

MaterCoeffGamma2 : double

Available since version 3.

CompParamBetaA : double

Available since version 3.

LatBuckParamBetaW : double

Available since version 3.

BendParamBetMy : double

Available since version 3.

BendParamBetaMz : double

Available since version 3.

BendParamBetaMlty : double

Available since version 3.

BendParamBetaMltz : double

Available since version 3.

MomStrengthMpzEuro : double

Available since version 3.

MaterialCapacityFy : double

Available since version 3.

ElastSectModulWyel : double

Available since version 3.

PlastSectModulWypl : double

Available since version 3.

ElastSectModulWzel : double

Available since version 3.

PlastSectModulWzpl : double

Available since version 3.

EffSectModulWyeff : double

Available since version 3.

EffSectModulWzeff : double

Available since version 3.

EffSectAreaSeff : double

Available since version 3.

FlangeAreaAf : double

Available since version 3.

FlangeAreaAw : double

Available since version 3.

AxForceExcentrEny : double

Available since version 3.

AxForcExcentrEnz : double

Available since version 3.

BuckCurveNumbY : double

Available since version 3.

BuckCurveNumbZ : double

Available since version 3.

BuckCurveCoeffAlfy : double

Available since version 3.

BuckCurveCoeffAlfz : double

Available since version 3.

BuckParamFiy : double

Available since version 3.

BuckParamFiz : double

Available since version 3.

MaEffRatio : double

Available since version 3.

PartEffRatio1 : double

Available since version 3.

PartEffRatio2 : double

Available since version 3.

PartEffRatio3 : double

Available since version 3.

PartEffRatio4 : double

Available since version 3.

PartEffRatio5 : double

Available since version 3.

IRDimYieldStrengthTypeEC3

Available since version 3.5.

Properties:

I_DYST_EC3_BASIC : = 0

Available since version 3.5.

I_DYST_EC3_AVERAGE : = 1

Available since version 3.5.

Code CB71

Available since version 3.5.

IRDimMembParamsCB71

Available since version 3.5.

Properties:

BucklingDiagramY : IRDimBuckDiagramCB71

Available since version 3.5.

BucklingDiagramZ : IRDimBuckDiagramCB71

Available since version 3.5.

BucklLengthCoeffY : double

Available since version 3.5.

BuckLengthCoeffZ : double

Available since version 3.5.

LatBuckType : IRDimLaterBuckTypeCB71

Available since version 3.5.

LatCoeffLowerFlangeValue : double

Available since version 3.5.

LatCoeffUpperFlangeValue : double

Available since version 3.5.

LoadLevel : IRDimLoadLevelCB71

Available since version 3.5.

LoadTypeY : IRDimLoadTypeCB71

Available since version 3.5.

TensAreaNetGros : double

Available since version 3.5.

TubeControl : bool

Available since version 3.5.

ArcBeamCheck : bool

Available since version 3.5.

ArcRadius : double

Available since version 3.5.

InflDeflCoefTheta : double

Available since version 3.5.

BuckIsComposedY : bool

Available since version 3.5.

BuckIsComposedZ : bool

Available since version 3.5.

BuckComposBeamCoefCey : double

Available since version 3.5.

BuckComposBeamCoefCez : double

Available since version 3.5.

HumChangeDeltaH : double

Available since version 3.5.

ForceDistFromBeginDistX : double

Available since version 3.5.

FireLftSideProt : bool

Available since version 3.5.

FireUprSideProt : bool

Available since version 3.5.

FireRgtSideProt : bool

Available since version 3.5.

FireLowSideProt : bool

Available since version 3.5.

FireProtectionTime : IRDimFireProtectionTimesCB71

Available since version 3.5.

FireMembPosition : IRDimFireMemberPositionsCB71

Available since version 3.5.

FireRequiredResist : double

Available since version 3.5.

FireContinProtForComposBeam : bool

Available since version 3.5.

IRDimCodeResCB71

Available since version 3.5.

Properties:

BadWidthBfUnderFire : bool

Available since version 3.5.

IsBuckY : short

Available since version 3.5.

IsBuckZ : short

Available since version 3.5.

StrsShearY : double

Available since version 3.5.

StrsShearZ : double

Available since version 3.5.

EfficiencyRatio : double

Available since version 3.5.

StrsFinal : double

Available since version 3.5.

StrsBend : double

Available since version 3.5.

StrsTens : double

Available since version 3.5.

StrsUprEdgeMY : double

Available since version 3.5.

StrsRgtEdgeMZ : double

Available since version 3.5.

StrsLowEdgeMY : double

Available since version 3.5.

StrsLftEdgeMZ : double
StrsBendInCompEdgeMY : double

Available since version 3.5.

StrsBendInCompEdgeMZ : double

Available since version 3.5.

StrsComp : double

Available since version 3.5.

LatBuckStrsCrit : double

Available since version 3.5.

StrsBendInCurveBeam : double

Available since version 3.5.

BuckLengthLfy : double

Available since version 3.5.

BuckLengthLfz : double

Available since version 3.5.

BuckLengthComposBeamLey : double

Available since version 3.5.

BuckLengthComposBeamLez : double

Available since version 3.5.

BuckSLendLay : double

Available since version 3.5.

BuckSlendLaz : double

Available since version 3.5.

BuckSlendComposBeamLaym : double

Available since version 3.5.

BuckSlendComposBeamLazm : double

Available since version 3.5.

LatBuckRelSlendLam : double

Available since version 3.5.

BuckCoefKy : double

Available since version 3.5.

BuckCoefKz : double

Available since version 3.5.

BuckCoefKMax : double

Available since version 3.5.

LatBuckCoefKinst : double

Available since version 3.5.

LatBuckLengthLd : double

Available since version 3.5.

LatBuckLengthCoef : double

Available since version 3.5.

TimberType : short

Available since version 3.5.

MatAxCompResist : double

Available since version 3.5.

MatAxTensResist : double

Available since version 3.5.

MatBendResist : double

Available since version 3.5.

MatShearResist : double

Available since version 3.5.

MatTrCompResist : double

Available since version 3.5.

MatTrTensResist : double

Available since version 3.5.

CoefDependOnAngle : double

Available since version 3.5.

CoefDependOnWidth : double

Available since version 3.5.

CoefDependOnHeight :

Available since version 3.5.

CoefDependOnHumAndComp : double

Available since version 3.5.

CoefDependOnHumAndBend : double

Available since version 3.5.

FireProtection : short

Available since version 3.5.

FireResistance : double

Available since version 3.5.

FireProtectionPosition : short

Available since version 3.5.

FireWidthB : double

Available since version 3.5.

FireWidthH : double

Available since version 3.5.

FireSecAreaS : double

Available since version 3.5.

FireShearAreaSY : double

Available since version 3.5.

FireShearAreaSZ : double

Available since version 3.5.

FireRgtEdgeDistVy : double

Available since version 3.5.

FireLftEdgeDistVpy : double

Available since version 3.5.

FireUprEdgeDistVz : double

Available since version 3.5.

FireLowEdgeDistVpz : double

Available since version 3.5.

FireInerMomIx : double

Available since version 3.5.

FireInerMomIy : double

Available since version 3.5.

FireInerMomIz : double

Available since version 3.5.

FireSeModulWy : double

Available since version 3.5.

FireSeModulWZ : double

Available since version 3.5.

FireRedCoefK1h : double

Available since version 3.5.

FireRedCoefK1b : double

Available since version 3.5.

FireProtectCoefK2 : double

Available since version 3.5.

FireWidthInMidSpanB : double

Available since version 3.5.

FireWidthAtMembEndB : double

Available since version 3.5.

FireHightInMidSpanH : double

Available since version 3.5.

FireWidthAtMembEndH : double

Available since version 3.5.

FireSecAreaInMidSpanS : double

Available since version 3.5.

FireInerMomInMidSpanIy : double

Available since version 3.5.

FireInerMomInMidSpanIz : double

Available since version 3.5.

IRDimBuckDiagramCB71

Available since version 3.5.

Properties:

I_BD_CB71_PINNED_PINNED_1_0 : = 0

Available since version 3.5.

I_BD_CB71_STIFF_STIFF_0_65 : = 1

Available since version 3.5.

I_BD_CB71_STIFF_PINNED_0_8 : = 2

Available since version 3.5.

I_BD_CB71_CANTILEVER : = 3

Available since version 3.5.

I_BD_CB71_NONE : = 4

Available since version 3.5.

I_BD_CB71_INTERNAL_BRACINGS : = 5

Available since version 3.5.

IRDimLaterBuckTypeCB71

Available since version 3.5.

Properties:

I_LBT_CB71_PINNED_SUPPORTS : = 0

Available since version 3.5.

I_LBT_CB71_FIXED_SUPPORTS : = 1

Available since version 3.5.

I_LBT_CB71_CANTILEVER : = 2

Available since version 3.5.

I_LBT_CB71_NONE : = 3

Available since version 3.5.

IRDimLoadLevelCB71

Available since version 3.5.

Properties:

I_LL_CB71_UP : = 0

Available since version 3.5.

I_LL_CB71_MIDDLE : = 1

Available since version 3.5.

I_LL_CB71_BOTTOM : = 2

Available since version 3.5.

IRDimLoadTypeCB71

Available since version 3.5.

Properties:

I_DLT_CB71_MOMENT_AT_END : = 0

Available since version 3.5.

I_DLT_CB71_UNIFORM_LOAD : = 1

Available since version 3.5.

I_DLT_CB71_FORCE_AT_DISTANCE : = 2

Available since version 3.5.

IRDimFireMemberPositionsCB71

Available since version 3.5.

Properties:

I_FMP_CB71_AUTOMATIC : = 0

Available since version 3.5.

I_FMP_CB71_VERTICAL : = 1

Available since version 3.5.

I_FMP_CB71_HORIZONTAL : = 2

Available since version 3.5.

IRDimFireProtectionTimesCB71

Available since version 3.5.

Properties:

I_FPT_CB71_LESS : = 0
< 1/4
Available since version 3.5.

I_FPT_CB71_MORE : = 1
> 1/4
Available since version 3.5.

Connection module

Names of interfaces defined for the connection module begin with the RJoint prefix.

Note:
All values described by interfaces of a connection module are given in millimeters.

IRJointConnectionDefType

A method of connection definition.

Properties:

I_JCDT_IN_STRUCTURE : = 0
a connection defined in a structure

I_JCDT_STANDALONE : = 1
a connection defined independently on a structure

IRJointConnectionType

Types of connections available in a connections module of Robot.

Properties:

I_JCT_KNEE_BOLTED : = 0
a bolted knee connection

I_JCT_KNEE_WELDED : = 1
a welded knee connection

I_JCT_ANGLE : = 2
angle connection

I_JCT_TRNS : = 3
transversal connection of beams (not available)

I_JCT_COL_PINNED : = 5
a pinned column base

I_JCT_COL_FIXED : = 6
a fixed column base

I_JCT_COL_CONCR : = 7
a concrete column base

I_JCT_TUBE : = 8
tube connection

I_JCT_UNKNOWN : = -1
I_JCT_GUSSET_SIMPLE : = 9
Available since version 1.7.

I_JCT_GUSSET_CROSS : = 10
Available since version 1.7.

I_JCT_GUSSET_FLANGE : = 11
Available since version 1.7.

I_JCT_BEAM_GIRDER : = 12

Available since version 7.5.

IRJointConnectionInfo

Information about connection.

Properties:

DefType : IRJointConnectionDefType
a method of connection definition

Elements : IRobotSelection
a selection of elements creating a connection

Node : long
number of a connection node

Number : long
connection number in the Robot connection server

Type : IRJointConnectionType
connection type

UniqueId : long
unique connection identifier
Available since version 1.7.

IRJointWebFlangeRelativePos

Possible types of a beam - column mutual position.

Properties:

I_JWFRP_WEB_LONG : = 0
I_JWFRP_WEB_TRAN : = 1
I_JWFRP_FLANGE_LONG : = 2
I_JWFRP_FLANGE_TRAN : = 3
I_JWFRP_OTHER : = 4

IRJointConnection

A data type describing common features for all connections that can be defined in Robot.

Properties:

WFRelPos : IRJointWebFlangeRelativePos
a type of relative position of a beam and a column

Type : IRJointConnectionType
a connection type

Operations:

SetToRobot (info : IRJointConnectionInfo)
The function sets and saves connection parameter in Robot.

GetFromRobot (info : IRJointConnectionInfo)
The function gets connection parameters connection parameters from Robot and fills an object with the data.

IRJointWeld

Description of a weld.

Properties:

Thick : double
weld thickness

Strength : double
weld resistance

IRJointPlate

Plate description.

Properties:

Length : double
plate length

Width : double
plate width

Thick : double
plate thickness

Exist : bool
information about a plate existance

IRJointBolts

Definition of a group of bolts with common parameters.

Properties:

Diameter : double
bolt diameter

ClassName : string
bolt class

DiameterName : string
bolt diameter name

Friction : double
bolt friction

Area : double
bolt section area

Rows : int
number of rows

Cols : int
number of columns

Height1 : double
distance of first bolt (e.g. from the top edge of a plate for knee connections)

IRJointBoltType

Available bolt types

Properties:

I_JBT_NORMAL : = 0
I_JBT_HIGH_TENSION : = 1

IRJointWebType

Types of reinforcing plates.

Properties:

I_JWT_SINGLE : = 0
I_JWT_DOUBLE : = 1

IRJointConnectionServer

The connection server administers all connection defined in the project. Each connection is associated with a number. The numbering of connections may be discontinuous.

Properties:

Count : long
number of defined connections

Operations:

CreateInfo () : IRJointConnectionInfo
The function creates and returns a new object representing the information about a connection.

Create (joint_type : IRJointConnectionType) : IRJointConnection
The function creates and returns an object representing a new connection of the indicated type.

GetInfo (num : long) : IRJointConnectionInfo
The function returns an object describing the connection with the given number.

Get (num : long) : IRJointConnection
The function returns the connection with the given number.

Delete (num : long)
The function removes the connection with the given number.

Find (sel_text : string) : long
The function finds a connection defined for the indicated selection of bars and returns its number. If there is no such connection, it returns the value 0.

GetAllNumbers () : IRobotNumbersArray
The function returns a table containing numbers of all defined connections.

Calculate (num : long, joint_loads : IRJointLoad) : double
The function carries out verification of the connection with the given number for the defined loads and returns the verification results.

CalculateNote (num : long, joint_loads : IRJointLoad, file_path : string) : double
The function carries out calculations for the indicated connection as the function Calculate(). Additionally, it generates a calculation note with calculation results.

Exist (num : long) : bool
The function checks if there is a connection with the indicated number.

FindWithId (unique_id : long) : long
Function returns number of a connection with the specified unique identifier. If the connection is not found, zero value is returned.

Available since version 1.7.

IRJointProfile

Structure describing a single connection component.

Properties:

Material : string
material name

Section : string
section name

Angle : double
connection angle

BarNumber : long
bar number
Available since version 3.5.

IRJointLoad

Interface describing a load applied to the connection.

Properties:

Cases : IRobotSelection
Selection of load cases, defined for the structure.

Type : IRJointLoadType
load type (defined manually or taken from Robot)

IRJointLoadType

Properties:

I_JLT_ROBOT_FORCES : = 1
load taken from structure calculations

I_JLT_MANUAL_FORCES : = 2
loads defined manually for the connection

Knee connection

IRJointKneeBolts

Derived from IRJointBolts
Definition of a bolt group for a knee connection.

Properties:

SpacingV : IRobotValuesArray
Distances between bolts in the vertical direction

SpacingH : IRobotValuesArray
Distances between bolts in the horizontal direction

IRJointKneeBracket

knee bracket

Properties:

ThickFlange : double
thickness of bracket flange

ThickWeb : double
thickness of bracket web

Length : double
bracket length

Width : double
bracket width

Height : double
bracket height

Angle : double
inclination angle of reinforcement

Exist : bool
information about a bracket existence

IRJointKneeWebPlate

Description of a plate stiffening a column web.

Properties:

Type : IRJointWebType
WeldTran : IRJointWeld

a transversal weld

WeldLong : IRJointWeld
a longitudinal weld

Bolts : IRJointKneeBolts
bolt group

Height : double
Available since version 1.7.

Width : double
Available since version 1.7.

Thick : double
Available since version 1.7.

IRJointKneeType

Accessible types of a knee connection.

Properties:

I_JKT_BEAM2BEAM : = 0
Beam-to-beam connection

I_JKT_BEAM2COLUMN : = 1
A frame knee connection.

I_JKT_BEAM2COLUMN_CONTINUE : = 2
A beam-to-column connection

IRJointKneeFixType

Available types of a knee connection as for joint type.

Properties:

I_JKFT_WELDED : = 0
I_JKFT_BOLTED : = 1

IRJointKneeReinfType

Available bracket types for a knee connection.

Properties:

I_JKRT_NONE : = 0
I_JKRT_STIFF : = 1

I_JKRT_BRACKET : = 2

IRJointKnee

Derived from IRJointConnection
Knee connection.

Properties:

KneeType : IRJointKneeType
a type of a knee connection

StiffTypeUp : IRJointKneeReinfType
a type of upper bracket

StiffTypeLow : IRJointKneeReinfType
a type of lower bracket

Bolts : IRJointKneeBolts
a group of connecting bolts

Plate : IRJointPlate
main plate

PlatePositionLow : double
distance between bottom plate edge and the bottom edge of the adjoining element

BracketUp : IRJointKneeBracket
upper bracket of a beam

BracketLow : IRJointKneeBracket
lower bracket of a beam

TensionPlateUp : IRJointPlate
top plate reinforcing the beam

TensionPlateLow : IRJointPlate
bottom plate reinforcing the beam

StiffUp : IRJointKneeStiffColumn
top column bracing

StiffLow : IRJointKneeStiffColumn
bottom column bracing

VStiffUp : IRJointPlate
a top, vertical bracing of a beam

VStiffLow : IRJointPlate
a bottom, vertical bracing of a beam

WeldWeb : IRJointWeld
WeldFlange : IRJointWeld
WeldStiff : IRJointWeld
WebStiffType : IRJointKneeWebStiffType
a bracket type of a column web

StiffDiag : IRJointKneeDiagonalStiff
inclined column bracket

WebPlate : IRJointKneeWebPlate
a plate reinforcing a column web

FixType : IRJointKneeFixType
out-of-date parameter - retained to ensure compatibility with the previous version

MaterPlates : IRobotMaterialData
Column : IRJointProfile
column section

Beam : IRJointProfile
beam section

IRJointKneeWebStiffType

Bracket types of a column web for a knee connection.

Properties:

I_JKWST_NONE : = 0
I_JKWST_PLATE : = 1
I_JKWST_DIAG : = 2

IRJointKneeDiagonalStiffType

Available slanting bracket types for a column web.

Properties:

I_JKDST_LEFT : = 0
I_JKDST_RIGHT : = 1
I_JKDST_DOUBLE : = 2

IRJointKneeDiagonalStiff

an inclined stiffener of a column.

Properties:

Type : IRJointKneeDiagonalStiffType
a stiffener type

Thick : double
thickness

Exist : bool

IRJointKneeStiffColumn

Column bracket in a knee connection.

Properties:

Thick : double
thickness

IRJointKneeLoad

Derived from IRJointLoad
Interface describing a load applied to a knee connection.

Properties:

M : double
N : double
Q : double

Footing connections

IRJointAnchorBolt

Anchorage bolt.

Properties:

Length1 : double
Available since version 1.7.

Length2 : double
Available since version 1.7.

Length3 : double
Available since version 1.7.

Length4 : double
Available since version 1.7.

IRJointAnchorPlate

Derived from IRJointPlate
Plate of an anchor bolt.

Properties:

Type : IRJointAnchorPlateType
shape of a plate near anchor
Available since version 1.7.

IRJointAnchorType

Available anchorage types.

Properties:

I_JAT_ROD : = 0
Available since version 1.7.

I_JAT_SIMPLE : = 1
Available since version 1.7.

I_JAT_PIN : = 2
Available since version 1.7.

I_JAT_PLATE : = 3
Available since version 1.7.

IRJointAnchor

Anchorage definition.

Properties:

Type : IRJointAnchorType
anchorage type
Available since version 1.7.

AnchorPlate : IRJointAnchorPlate
plate of an anchor bolt
Available since version 1.7.

Tige : IRJointAnchorBolt
parameters of anchor bolt
Available since version 1.7.

Bolts : IRJointFootBolts
arrangement and type of anchor bolts

Available since version 1.7.

IRJointFootBolts

Bolt group of identical parameters for pinned column base.

Properties:

SpacingH : double
Available since version 1.7.

SpacingV : double
Available since version 1.7.

Diameter : long
bolt diameter
Available since version 1.7.

ClassName : string
name of the bolt class
Available since version 1.7.

DiameterName : string
Available since version 1.7.

Friction : double
Available since version 1.7.

Area : double
Available since version 1.7.

Rows : long
Available since version 1.7.

Distance : double
Available since version 1.7.

IRJointFootPlateType

Available connection types - plate/angle.

Properties:

I_JFPT_NONE : = 0
Available since version 1.7.

I_JFPT_WELD : = 1
Available since version 1.7.

I_JFPT_ANGLE : = 2
Available since version 1.7.

IRJointFootPlate

Derived from IRJointPlate

Properties:

Type : IRJointFootPlateType
connection type - plate/angle
Available since version 1.7.

Diameter : double
opening diameters in a plate for bolts
Available since version 1.7.

IRJointWedge

Properties:

Type : IRJointWedgeType
wedge type
Available since version 1.7.

Length : double
Available since version 1.7.

Profile : IRJointProfile
wedge section
Available since version 1.7.

Width : double
Available since version 1.7.

Thick : double
Available since version 1.7.

XTypeMaterial : string
Available since version 1.7.

IRJointWedgeType

Available wedge types.

Properties:

I_JWT_NONE : = 0
Available since version 1.7.

I_JWT_I : = 1
Available since version 1.7.

I_JWT_L : = 2
Available since version 1.7.

I_JWT_T : = 3
Available since version 1.7.

I_JWT_X : = 4
Available since version 1.7.

IRJointBearingPlate

Derived from IRJointPlate

Properties:

ThickBearingBar : double
thickness of limit element
Available since version 1.7.

IRJointFootWelds

Description of connection element welds.

Properties:

FootPlate : IRJointWeld
Available since version 1.7.

Wedge : IRJointWeld
Available since version 1.7.

Washer : IRJointWeld
Available since version 1.7.

Bearing : IRJointWeld
Available since version 1.7.

Stiff : IRJointWeld
Available since version 1.7.

IRJointColumnBracket

Column stiffener

Properties:

Thick : double
Available since version 1.7.

Width : double
Available since version 1.7.

Length : double
Available since version 1.7.

Height : double
Available since version 1.7.

VThick : double
Available since version 1.7.

Operations:

Exist () : bool
Function returns information about the stiffener existence.
Available since version 1.7.

IRJointAnchorPlateType

Types of plates near anchor.

Properties:

I_JAPT_RECT : = 0
Available since version 1.7.

I_JAPT_TUBE : = 1
Available since version 1.7.

IRJointFootStiffenerVert

Properties:

Thick : double
Available since version 1.7.

Length : double
Available since version 1.7.

WidthSpacing : double
Available since version 1.7.

IRJointFootStiffenerHoriz

Properties:

Thick : double
Available since version 1.7.

IRJointFootMaterials

Properties:

PlateSigma : double
Available since version 1.7.

PlateYoung : double
Available since version 1.7.

CementAmount : double
Available since version 1.7.

ConcrSteelCoeff : double
Available since version 1.7.

ConcrClass : double
Available since version 1.7.

IRJointColumnSquare

Properties:

Profile : IRJointProfile
Available since version 1.7.

Length : double
Available since version 1.7.

Diameter : double
Available since version 1.7.

IRJointFootStiffenerSimple

Properties:

Length : double
Available since version 1.7.

Height : double
Available since version 1.7.

Width : double
Available since version 1.7.

Thick : double
Available since version 1.7.

Type : IRJointFootStiffType
Available since version 1.7.

IRJointFootStiffenerComplex

Properties:

Length : double
Available since version 1.7.

Height : double
Available since version 1.7.

ThickPlateHor : double
Available since version 1.7.

ThickStiff : double
Available since version 1.7.

IRJointFootStiffType

Properties:

I_JFST_1 : = 1
Available since version 1.7.

I_JFST_2 : = 2
Available since version 1.7.

I_JFST_3 : = 3
Available since version 1.7.

I_JFST_4 : = 4
Available since version 1.7.

I_JFST_5 : = 5
Available since version 1.7.

Pinned column base

IRJointPinnedColumnBaseStiffType

Properties:

I_JPCBST_LRDIAG : = 0
Available since version 1.7.

I_JPCBST_LDIAG : = 1
Available since version 1.7.

I_JPCBST_RDIAG : = 2
Available since version 1.7.

I_JPCBST_LRI : = 3
Available since version 1.7.

I_JPCBST_LI : = 4
Available since version 1.7.

I_JPCBST_RI : = 5
Available since version 1.7.

I_JPCBST_NONE : = 6
Available since version 1.7.

IRJointPinnedColumnBase

Derived from IRJointConnection
Pinned column base

Properties:

StiffType : IRJointPinnedColumnBaseStiffType
Available since version 1.7.

Profile : IRJointProfile
column section
Available since version 1.7.

Bolts : IRJointFootBolts
group of connecting bolts
Available since version 1.7.

Anchor : IRJointAnchor
anchorage
Available since version 1.7.

FootPlate : IRJointFootPlate
plate
Available since version 1.7.

Wedge : IRJointWedge
wedge
Available since version 1.7.

Base : IRJointPlate
base
Available since version 1.7.

Washer : IRJointPlate
washer
Available since version 1.7.

Bearing : IRJointPlate
bearing
Available since version 1.7.

BearingPlate : IRJointBearingPlate
bearing plate
Available since version 1.7.

Welds : IRJointFootWelds
welds in a connection
Available since version 1.7.

StiffHoriz : IRJointFootStiffenerHoriz
Available since version 1.7.

StiffVert : IRJointFootStiffenerVert
Available since version 1.7.

Materials : IRJointFootMaterials
Available since version 1.7.

Square : IRJointColumnSquare
Available since version 1.7.

BasePlateMaterial : IRobotMaterialData

Available since version 4.5.

NodeNumber : long

Available since version 4.5.

IRJointPinnedLoad

Derived from IRJointLoad

Properties:

Nt : double
Available since version 1.7.

Nc : double
Available since version 1.7.

Tz : double
Available since version 1.7.

NTz : double
Available since version 1.7.

Ty : double
Available since version 1.7.

NTy : double
Available since version 1.7.

Fixed spread footing

IRJointFixedLoad

Derived from IRJointLoad

Properties:

Fx : double
Available since version 1.7.

Fy : double
Available since version 1.7.

Fz : double
Available since version 1.7.

My : double

Available since version 1.7.

Mz : double

Available since version 7.5.

IRJointFixedColumnBase

Derived from IRJointConnection

Properties:

StiffType : IRJointFixedColumnBaseStiffType
Available since version 1.7.

Bolts : IRJointFootBolts
Available since version 1.7.

Anchor : IRJointAnchor
Available since version 1.7.

FootPlate : IRJointFootPlate
Available since version 1.7.

Wedge : IRJointWedge
Available since version 1.7.

Base : IRJointPlate
Available since version 1.7.

Washer : IRJointPlate
Available since version 1.7.

Welds : IRJointFixedFootWelds
Available since version 1.7.

SimpleStiff : IRJointFootStiffenerSimple
Available since version 1.7.

ComplexStiff : IRJointFootStiffenerComplex
Available since version 1.7.

Profile : IRJointProfile
Available since version 1.7.

Materials : IRJointFootMaterials
Available since version 1.7.

BasePlateMaterial : IRobotMaterialData

Available since version 4.5.

NodeNumber : long

Available since version 4.5.

IRJointFixedFootWelds

Properties:

FootPlate : IRJointWeld
Available since version 1.7.

Wedge : IRJointWeld
Available since version 1.7.

Washer : IRJointWeld
Available since version 1.7.

Stiff : IRJointWeld
Available since version 1.7.

IRJointFixedColumnBaseStiffType

Properties:

I_JFCBST_NONE : = 0
Available since version 1.7.

I_JFCBST_SIMPLE : = 1
Available since version 1.7.

I_JFCBST_COMPLEX : = 2
Available since version 1.7.

Spread footing

IRJointConcreteMaterials

Properties:

Sigma : double
Available since version 1.7.

Dosage : double
Available since version 1.7.

CoeffBA : double
Available since version 1.7.

IRJointConcreteColumn

Derived from IRJointConnection

Properties:

SpreadFootingType : IRJointSpreadFootingType
Profile : IRJointProfile
Depth : double
Materials : IRJointConcreteMaterials

Available since version 7.5.

Base : IRJointPlate

Available since version 7.5.

Foundation : IRJointConcreteColumnFoundation

Available since version 7.5.

CalcModel : IRJointColumnBasePlateCalcModel

Available since version 7.5.

IRJointSpreadFootingType

Available since version 7.5.

Properties:

I_JSFT_RECT_CROSS_SECT : = 1

Available since version 7.5.

I_JSFT_ABRUPT_CHANGES_HEIGHT : = 2

Available since version 7.5.

I_JSFT_TAPERED : = 3

Available since version 7.5.

IRJointConcreteColumnFoundation

Available since version 7.5.

Properties:

par_a1 : double

Available since version 7.5.

par_l : double

Available since version 7.5.

par_l1 : double

Available since version 7.5.

par_l2 : double

Available since version 7.5.

par_b : double

Available since version 7.5.

par_b1 : double

Available since version 7.5.

par_b2 : double

Available since version 7.5.

par_h : double

Available since version 7.5.

par_h1 : double

Available since version 7.5.

par_h2 : double

Available since version 7.5.

par_o1 : double

Available since version 7.5.

par_o2 : double

Available since version 7.5.

IRJointColumnBasePlateCalcModel

Available since version 7.5.

Properties:

I_JCBPCM_PLASTIC : = 0

Available since version 7.5.

I_JCBPCM_ELASTIC : = 1

Available since version 7.5.

IRJointConcreteColumnLoad

Derived from IRJointLoad

Available since version 7.5.

Properties:

Fx : double

Available since version 7.5.

Fy : double

Available since version 7.5.

Fz : double

Available since version 7.5.

My : double

Available since version 7.5.

Mz : double

Available since version 7.5.

FxAssemb : double

Available since version 7.5.

Angle connection

IRJointAngleBolts

Properties:

Diameter : long
Available since version 1.7.

ClassName : string
Available since version 1.7.

DiameterName : string
Available since version 1.7.

Friction : double
Available since version 1.7.

Area : double
Available since version 1.7.

Rows : long
Available since version 1.7.

DistFromUpperEdge : double
Available since version 1.7.

DistFromWeb : double
Available since version 1.7.

Spacing : double
Available since version 1.7.

IRJointAngle

Properties:

Type : IRJointAngleType
Available since version 1.7.

Profile : IRJointProfile
Available since version 1.7.

Length : double
angle height
Available since version 1.7.

DistFromUpperBeamEdge : double
distance from the top beam edge
Available since version 1.7.

ProfilePosition : IRJointAngleProfilePosition
Available since version 1.7.

IRJointAngleType

Properties:

I_JST_ALONG_BEARER : = 0
Available since version 1.7.

I_JST_ALONG_SECOND : = 1
Available since version 1.7.

IRJointBeamCut

Definition of beam cut-outs.

Properties:

Length : double
cut-out length
Available since version 1.7.

HeightUpper : double
length of a beam cut-out (top)
Available since version 1.7.

HeightLower : double
length of a beam cut-out (bottom)
Available since version 1.7.

IRJointWithAngles

Derived from IRJointConnection
Angle connection

Properties:

BeamBolts : IRJointAngleBolts
bolt group on the beam side
Available since version 1.7.

ColumnBolts : IRJointAngleBolts
bolt group on the column side
Available since version 1.7.

Distance : double
distance between a column and beam
Available since version 1.7.

BeamCut : IRJointBeamCut
beam cut-out
Available since version 1.7.

Angle : IRJointAngle
angle
Available since version 1.7.

BeamProfile : IRJointProfile
beam section
Available since version 1.7.

ColumnProfile : IRJointProfile
column section
Available since version 1.7.

IRJointAngleProfilePosition

Properties:

I_JAPP_LONG_FLANGE_TO_COLUMN : = 0
Available since version 1.7.

I_JAPP_SHORT_FLANGE_TO_COLUMN : = 1
Available since version 1.7.

IRJointAngleLoad

Derived from IRJointLoad

Properties:

Fz : double
Available since version 1.7.

Tube connection

IRJointTube

Derived from IRJointConnection

Properties:

TubeType : IRJointTubeType
type of tube connection
Available since version 1.7.

Flange : IRJointTubeFlangeProfile
chord section
Available since version 1.7.

DiagLeftUpper : IRJointTubeDiagProfile
section of a first diagonal
Available since version 1.7.

DiagRightUpper : IRJointTubeDiagProfile
Available since version 1.7.

DiagLeftLower : IRJointTubeDiagProfile
Available since version 1.7.

PostUpper : IRJointTubePostProfile
Available since version 1.7.

StiffHoriz : IRJointPlate
Available since version 1.7.

StiffLateral : IRJointPlate
Available since version 1.7.

WeldDiag : IRJointWeld
Available since version 1.7.

WeldStiff : IRJointWeld
Available since version 1.7.

IRJointTubeType

Properties:

I_JTT_T : = 0
Available since version 1.7.

I_JTT_Y : = 1
Available since version 1.7.

I_JTT_X : = 2
Available since version 1.7.

I_JTT_N : = 3
Available since version 1.7.

I_JTT_K : = 4
Available since version 1.7.

I_JTT_KT : = 5
Available since version 1.7.

I_JTT_NONE : = -1
Available since version 1.7.

IRJointTubeProfileType

Properties:

I_JTPT_RECT : = 0
Available since version 1.7.

I_JTPT_TUBE : = 1
Available since version 1.7.

I_JTPT_RECT_RECT : = 2
Available since version 1.7.

I_JTPT_RECT_TUBE : = 3
Available since version 1.7.

I_JTPT_TUBE_TUBE : = 4
Available since version 1.7.

I_JTPT_UNKNOWN : = -1
Available since version 1.7.

IRJointTubeFlangeProfile

Properties:

Section : string
Available since version 1.7.

Excentr : double
Available since version 1.7.

Material : string
Available since version 1.7.

BarNumber : long
bar number
Available since version 3.5.

Exist : bool

Available since version 4.5.

IRJointTubeDiagProfile

Properties:

Section : string
Available since version 1.7.

Length : double
Available since version 1.7.

Angle : double
Available since version 1.7.

Gap : double
Available since version 1.7.

Overlap : double
Available since version 1.7.

Material : string
Available since version 1.7.

BarNumber : long
bar number
Available since version 3.5.

Exist : bool

Available since version 4.5.

IRJointTubePostProfile

Properties:

Section : string
Available since version 1.7.

Length : double
Available since version 1.7.

Material : string
Available since version 1.7.

Exist : bool

Available since version 4.5.

BarNumber : long

Available since version 4.5.

IRJointTubeLoad

Derived from IRJointLoad

Properties:

FlangeN : double
Available since version 1.7.

FlangeQ : double
Available since version 1.7.

FlangeM : double
Available since version 1.7.

DiagLeftUpperN : double
Available since version 1.7.

DiagLeftUpperM : double
Available since version 1.7.

DiagRightUpperN : double
Available since version 1.7.

DiagRightUpperM : double
Available since version 1.7.

PostUpperN : double
Available since version 1.7.

DiagLeftLowerN : double

Available since version 1.7.

DiagLeftLowerM : double
Available since version 1.7.

Gusset plate

IRJointGussetBoltsDiag

Properties:

Diameter : double
Available since version 1.7.

ClassName : string
Available since version 1.7.

Friction : double
Available since version 1.7.

Rows : long
Available since version 1.7.

Spacing : IRobotValuesArray
Available since version 1.7.

DistanceH1 : double
Available since version 1.7.

BoltAxisShift : double
Available since version 1.7.

IRJointGussetWeldsDiag

Properties:

ThickFlange : double
Available since version 1.7.

ThickCorner : double
Available since version 1.7.

IRJointGussetDiagonale

Properties:

Profile : IRJointProfile
Available since version 1.7.

BoltsDiag : IRJointGussetBoltsDiag
Available since version 1.7.

WeldsDiag : IRJointGussetWeldsDiag
Available since version 1.7.

DistanceEC : double
Available since version 1.7.

Position : IRJointGussetDiagonalePositionType
Available since version 1.7.

Exist : bool

Available since version 4.5.

BarNumber : long

Available since version 4.5.

IRJointGussetFixType

Properties:

I_JGFT_BOLTED : = 0
Available since version 1.7.

I_JGFT_WELDED : = 1
Available since version 1.7.

IRJointGussetSimpleProfilePosition

Properties:

I_JGSPP_VERTICAL : = 0
Available since version 1.7.

I_JGSPP_DIAGONAL : = 1
Available since version 1.7.

IRJointGussetSimple

Derived from IRJointConnection

Properties:

ProfilePosition : IRJointGussetSimpleProfilePosition
Available since version 1.7.

FixType : IRJointGussetFixType
Available since version 1.7.

Diagonale : IRJointGussetDiagonale
Available since version 1.7.

GussetPlate : IRJointGussetSimplePlate
Available since version 1.7.

Attachment : IRJointGussetSimpleAttachment
Available since version 1.7.

IRJointGussetSimplePlate

Properties:

Width : double
Available since version 1.7.

Height : double
Available since version 1.7.

Thick : double
Available since version 1.7.

DistanceEH : double
Available since version 1.7.

DistanceEV : double
Available since version 1.7.

DistanceD : double
Available since version 1.7.

Material : string
Available since version 1.7.

CornersType : IRJointGussetCornersType
Available since version 1.7.

IRJointGussetCornersType

Properties:

I_JGCT_ACUTE : = 0
Available since version 1.7.

I_JGCT_CUT_IN : = 1
Available since version 1.7.

I_JGCT_CUT_OUT : = 2
Available since version 1.7.

IRJointGussetSimpleAttachment

Properties:

FixType : IRJointGussetFixType
Available since version 1.7.

BoltsAttach : IRJointGussetSimpleAttachBolts
Available since version 1.7.

WeldsAttach : IRJointGussetSimpleAttachWelds
Available since version 1.7.

IRJointGussetSimpleAttachBolts

Properties:

Diameter : double
Available since version 1.7.

ClassName : string
Available since version 1.7.

Friction : double
Available since version 1.7.

Horizontal : IRJointGussetSimpleAttachBoltsHorizontal
Available since version 1.7.

Vertical : IRJointGussetSimpleAttachBoltsVertical
Available since version 1.7.

IRJointGussetSimpleAttachBoltsHorizontal

Properties:

Rows : long
Available since version 1.7.

Spacing : IRobotValuesArray
Available since version 1.7.

DistanceH1 : double
Available since version 1.7.

DistanceEB : double
Available since version 1.7.

IRJointGussetSimpleAttachBoltsVertical

Properties:

Rows : long
Available since version 1.7.

Spacing : IRobotValuesArray
Available since version 1.7.

DistanceH1 : double
Available since version 1.7.

DistanceEB : double
Available since version 1.7.

IRJointGussetSimpleAttachWelds

Properties:

ThickEdgeB : double
Available since version 1.7.

ThickEdgeA : double
Available since version 1.7.

IRJointGussetCrossProfileCutting

Properties:

I_JGCPC_CONTINUE_LEFTUPPER_RIGHTLOWER : = 0
Available since version 1.7.

I_JGCPC_CONTINUE_LEFTLOWER_RIGHTUPPER : = 1
Available since version 1.7.

I_JGCPC_ALL_CUTTED : = 2
Available since version 1.7.

IRJointGussetCrossPlate

Properties:

B1 : double
Available since version 1.7.

B2 : double
Available since version 1.7.

B3 : double
Available since version 1.7.

B4 : double
Available since version 1.7.

H1 : double
Available since version 1.7.

H2 : double
Available since version 1.7.

H3 : double
Available since version 1.7.

H4 : double
Available since version 1.7.

Thick : double
Available since version 1.7.

Material : string
Available since version 1.7.

CornersType : IRJointGussetCornersType
Available since version 1.7.

IRJointGussetCross

Derived from IRJointConnection

Properties:

ProfileCutting : IRJointGussetCrossProfileCutting
Available since version 1.7.

FixType_LeftU_RightL : IRJointGussetFixType
Available since version 1.7.

FixType_LeftL_RightU : IRJointGussetFixType
Available since version 1.7.

DiagLeftUpper : IRJointGussetDiagonale
Available since version 1.7.

DiagLeftLower : IRJointGussetDiagonale
Available since version 1.7.

DiagRightUpper : IRJointGussetDiagonale
Available since version 1.7.

DiagRightLower : IRJointGussetDiagonale
Available since version 1.7.

GussetPlate : IRJointGussetCrossPlate
Available since version 1.7.

IRJointGussetFlangeProfileCutting

Properties:

I_JGFPC_CONTINUOUS : = 0
Available since version 1.7.

I_JGFPC_CUTTED : = 1
Available since version 1.7.

IRJointGussetFlangePlate

Properties:

B1 : double
Available since version 1.7.

B2 : double
Available since version 1.7.

B3 : double
Available since version 1.7.

B4 : double
Available since version 1.7.

H1 : double
Available since version 1.7.

H2 : double
Available since version 1.7.

H3 : double
Available since version 1.7.

H4 : double
Available since version 1.7.

EH : double
Available since version 1.7.

EV : double
Available since version 1.7.

Thick : double
Available since version 1.7.

Material : string
Available since version 1.7.

CornersType : IRJointGussetCornersType
Available since version 1.7.

Type : IRJointGussetFlangePlateRegularType
Available since version 1.7.

IRJointGussetFlange

Derived from IRJointConnection

Properties:

FixType_FlangeLeft : IRJointGussetFixType
Available since version 1.7.

FixType_FlangeRight : IRJointGussetFixType
Available since version 1.7.

FixType_DiagonalLeft : IRJointGussetFixType
Available since version 1.7.

FixType_DiagonalRight : IRJointGussetFixType
Available since version 1.7.

FixType_DiagonalUp : IRJointGussetFixType
Available since version 1.7.

ProfileCutting : IRJointGussetFlangeProfileCutting
Available since version 1.7.

FlangeLeft : IRJointGussetDiagonale
Available since version 1.7.

FlangeRight : IRJointGussetDiagonale
Available since version 1.7.

DiagonalLeft : IRJointGussetDiagonale
Available since version 1.7.

DiagonalRight : IRJointGussetDiagonale
Available since version 1.7.

DiagonalUp : IRJointGussetDiagonale
Available since version 1.7.

GussetPlate : IRJointGussetFlangePlate
Available since version 1.7.

IRJointGussetSimpleLoad

Derived from IRJointLoad

Properties:

Fx : double
Available since version 1.7.

IRJointGussetCrossLoad

Derived from IRJointLoad

Properties:

Fx_LeftUpper : double
Available since version 1.7.

Fx_LeftLower : double
Available since version 1.7.

Fx_RightUpper : double
Available since version 1.7.

Fx_RightLower : double
Available since version 1.7.

IRJointGussetFlangeLoad

Derived from IRJointLoad

Properties:

Fx_FlangeLeft : double
Available since version 1.7.

Fx_FlangeRight : double
Available since version 1.7.

Fx_LeftUpper : double
Available since version 1.7.

Fx_Upper : double
Available since version 1.7.

Fx_RightUpper : double
Available since version 1.7.

IRJointGussetDiagonalePositionType

Properties:

I_JGDPT_LONG_FLANGE : = 0
Available since version 1.7.

I_JGDPT_SHORT_FLANGE : = 1
Available since version 1.7.

I_JGDPT_LONG_FLANGE_INV : = 2
Available since version 1.7.

I_JGDPT_SHORT_FLANGE_INV : = 3
Available since version 1.7.

IRJointGussetFlangePlateRegularType

Properties:

I_JGFPRT_REGULAR : = 0
Available since version 1.7.

I_JGFPRT_IRREGULAR : = 1
Available since version 1.7.

Available since version 7.5.

IRJointBeamGirderBolts

Bolt description for stiffener and plate
Available since version 7.5.

Properties:

Diameter : long
Bolt diameter
Available since version 7.5.

ClassName : string
Bolt class
Available since version 7.5.

DiameterName : string

Available since version 7.5.

Friction : double

Available since version 7.5.

Area : double

Available since version 7.5.

Rows : long
Number of bolt rows
Available since version 7.5.

Cols : long

Available since version 7.5.

SpacingV : double
vertical spacing
Available since version 7.5.

SpacingH : double
horizontal spacing
Available since version 7.5.

DistFromUpperBeamEdge : double
Distance of 1st bolt row to upper beam edge
Available since version 7.5.

DistFromVertBeamEdge : double
Distance of 1st bolt column to vertical beam edge
Available since version 7.5.

IRJointBeamGirderStiffener

Stiffener description
Available since version 7.5.

Properties:

Length : double
Stiffener length
Available since version 7.5.

Thick : double
Stiffener thickness
Available since version 7.5.

CutLength : double
length of stiffener cut-off
Available since version 7.5.

CutHeightUp : double

Available since version 7.5.

CutHeightDown : double

Available since version 7.5.

Material : string
Stiffener material
Available since version 7.5.

ConnectorsType : IRJointConnectorsType
connector type for stiffener-beam connection

Available since version 7.5.

BoltsBeam : IRJointBeamGirderBolts
bolts for stiffener-beam connection
Available since version 7.5.

WeldsBeam : IRJointWeld
welds for stiffener-beam connection
Available since version 7.5.

IRJointBeamGirderPlate

Plate description
Available since version 7.5.

Properties:

Length : double
Plate length
Available since version 7.5.

Width : double
Plate width
Available since version 7.5.

Thick : double
Plate thickness
Available since version 7.5.

DistFromUpperBeamEdge : double
distance of upper plate edge to upper beam edge
Available since version 7.5.

Material : string
Material name for plate
Available since version 7.5.

ConnectorsType : IRJointConnectorsType
connector type for the plate-beam connection
Available since version 7.5.

BoltsBeam : IRJointBeamGirderBolts
bolts connecting plate with beam
Available since version 7.5.

WeldsGirder : IRJointWeld
welds connecting principal beam flange with plate
Available since version 7.5.

WeldsBeam : IRJointWeld

welds connecting plate with beam
Available since version 7.5.

IRJointBeamGirderSeat

Description of seat profile
Available since version 7.5.

Properties:

Section : string
section name
Available since version 7.5.

Material : string
name of seat profile material
Available since version 7.5.

Length : double
length of seat profile
Available since version 7.5.

ConnectorsToWebType : IRJointConnectorsType
connector type for seat profile-principal beam web connection
Available since version 7.5.

BoltsGirder : IRJointBeamGirderSeatBolts
bolts connecting seat profile and principal beam web
Available since version 7.5.

BoltsBeam : IRJointBeamGirderSeatBolts
bolts connecting seat profile and beam flange
Available since version 7.5.

WeldsGirder : IRJointWeld
welds connecting seat profile and principal beam web
Available since version 7.5.

IRJointBeamGirder

Derived from IRJointConnection
Beam-principal beam connection
Available since version 7.5.

Properties:

Girder : IRJointProfile
Section of principal beam
Available since version 7.5.

BeamLeft : IRJointBeamGirderBeam
Right beam section
Available since version 7.5.

BeamRight : IRJointBeamGirderBeam
Section of principal beam
Available since version 7.5.

AngleLeft : IRJointBeamGirderAngle

Available since version 7.5.

AngleRight : IRJointBeamGirderAngle

Available since version 7.5.

StiffenerLeft : IRJointBeamGirderStiffener

Available since version 7.5.

StiffenerRight : IRJointBeamGirderStiffener

Available since version 7.5.

PlateLeft : IRJointBeamGirderPlate

Available since version 7.5.

PlateRight : IRJointBeamGirderPlate

Available since version 7.5.

SeatLeftUp : IRJointBeamGirderSeat

Available since version 7.5.

SeatLeftDown : IRJointBeamGirderSeat

Available since version 7.5.

SeatRightUp : IRJointBeamGirderSeat

Available since version 7.5.

SeatRightDown : IRJointBeamGirderSeat

Available since version 7.5.

DistGirderToBeamLeft : double

Available since version 7.5.

DistGirderToBeamRight : double

Available since version 7.5.

DistGirderFlangeToBeamFlangeLeft : double

Available since version 7.5.

DistGirderFlangeToBeamFlangeRight : double

Available since version 7.5.

ElementType : IRJointElementType

Available since version 7.5.

IRJointBeamGirderLoad

Derived from IRJointLoad

Available since version 7.5.

Properties:

LFx : double
Axial force (left side)
Available since version 7.5.

LFz : double
Shear force (left side)
Available since version 7.5.

LMy : double
Bending moment (left side)
Available since version 7.5.

RFx : double
Axial force (right side)
Available since version 7.5.

RFz : double
Shear force (right side)
Available since version 7.5.

RMy : double
Bending moment (right side)
Available since version 7.5.

IRJointBeamGirderSeatBolts

Available since version 7.5.

Properties:

Diameter : long

Available since version 7.5.

ClassName : string

Available since version 7.5.

DiameterName : string

Available since version 7.5.

Friction : double

Available since version 7.5.

Area : double

Available since version 7.5.

Cols : long

Available since version 7.5.

DistFromVertEdge : double
Distance of extreme bolt to vertical edge of seat profile
Available since version 7.5.

DistFromPerpendicularArmEdge : double
Distance of bolt row to edge of perpendicular leg of seat profile
Available since version 7.5.

IRJointConnectorsType

Available since version 7.5.

Properties:

I_JCT_BOLTS : = 0

Available since version 7.5.

I_JCT_WELDS : = 1

Available since version 7.5.

IRJointElementType

Available since version 7.5.

Properties:

I_JET_R_ANGLES_WEB : = 0

Available since version 7.5.

I_JET_R_ANGLES_FLANGE : = 1

Available since version 7.5.

I_JET_R_STIFF_SHORT : = 2

Available since version 7.5.

I_JET_R_STIFF_LONG : = 3

Available since version 7.5.

I_JET_R_SEATS : = 4

Available since version 7.5.

I_JET_R_SEATS_ANGLES : = 5

Available since version 7.5.

I_JET_R_PLATES : = 6

Available since version 7.5.

I_JET_L_ANGLES_R_ANGLES_WEB : = 7

Available since version 7.5.

I_JET_L_ANGLES_R_ANGLES_FLANGE : = 8

Available since version 7.5.

I_JET_L_STIFF_SHORT_R_STIFF_SHORT : = 9

Available since version 7.5.

I_JET_L_STIFF_SHORT_R_STIFF_LONG : = 10

Available since version 7.5.

I_JET_L_STIFF_LONG_R_STIFF_SHORT : = 11

Available since version 7.5.

I_JET_L_STIFF_LONG_R_STIFF_LONG : = 12

Available since version 7.5.

I_JET_L_SEATS_R_SEATS : = 13

Available since version 7.5.

I_JET_L_SEATS_R_SEATS_ANGLES : = 14

Available since version 7.5.

I_JET_L_SEATS_ANGLES_R_SEATS : = 15

Available since version 7.5.

I_JET_L_SEATS_ANGLES_R_SEATS_ANGLES : = 16

Available since version 7.5.

I_JET_L_PLATES_R_PLATES : = 17

Available since version 7.5.

I_JET_L_PLATES_R_ANGLES : = 18

Available since version 7.5.

I_JET_L_ANGLES_R_PLATES : = 19

Available since version 7.5.

IRJointBeamGirderAngle

Available since version 7.5.

Properties:

Element : IRJointAngle

Available since version 7.5.

BoltsGirder : IRJointAngleBolts

Available since version 7.5.

BoltsBeam : IRJointAngleBolts

Available since version 7.5.

IRJointBeamGirderBeam

Available since version 7.5.

Properties:

Profile : IRJointProfile

Available since version 7.5.

CutEnd : IRJointBeamCut

Available since version 7.5.

Correction of the structure

Available since version 3.5.

IRobotBarIntersectRelationship

Relations between bars resulting from bar intersections.
Available since version 3.5.

Properties:

I_BIR_UNKNOWN : = 0
it is not known if bars remaining in this relation intersect or not
Available since version 3.5.

I_BIR_NOT_INTERSECT : = 1
bars remaining in this relation do not intersect
Available since version 3.5.

I_BIR_INTERSECT_MID_MID : = 2
bars remaining in this relation intersect
Available since version 3.5.

I_BIR_INTERSECT_MID_END : = 3

Available since version 3.5.

I_BIR_INTERSECT_END_MID : = 4

Available since version 3.5.

I_BIR_INTERSECT_END_END : = 5

Available since version 3.5.

IRobotStructureGeoAnalyser

Tool component performing anlysis of structure geometry and its automatic correction in order to obtain a correct calculation model.
Available since version 3.5.

Properties:

Precision : double
precision applied by the Correct function;
setting the value (-1) indicates adoption of a default value
Available since version 3.5.

CanExtendBars : bool
flag managing operation of the Correct function;
True (default) value means that to carry out structure correction bar length may be increased
Available since version 3.5.

CanUseRigidLinks : bool
flag managing operation of the Correct function;
True (default) value means that to carry out structure correction rigid links may be defined
Available since version 3.5.

Operations:

StartCollectInfo ()
Function runs analysis of operations performed on the structure to make subsequent correction of its geometry easier.
Available since version 3.5.

StopCollectInfo ()

Available since version 3.5.

DefineIntersection (_bar1 : long, _bar2 : long, _intersect_rel : IRobotBarIntersectRelationship)
Function defines a relation of intersection between the specified bars. This information will be used by the function of structure geometry correction.
Available since version 3.5.

Correct (_bar_sel : string = "")
Function carries out geometry correction for a whole structure or for a structure part based on the parameters set earlier.
Available since version 3.5.

SetNodeSize (_node_num : long, _radius : double)
Function defines the "size" of a node. Structure elements that are located in a distance equal or less than the specified radius from the node will be treated as adjoining to the node.

SetEdgeSize (_obj_num : long, _radius : double)
Function defines the "thickness" of a geometric object edge.

STR files analyser

Available since version 2.5.

IRobotSTRFileAnalyser

Analyzer of STR format text files.
Available since version 2.5.

Properties:

InsertParams : IRobotFileInsertParams
parameters of inserting the structure defined in an STR file to the current project
Available since version 2.5.

Params : IRobotSTRParams
parameters defined in the STR file
Available since version 2.5.

Operations:

ReadParams (_file_path : string) : bool
Function reads in the parameters defined in the indicated STR file. Once read in, the parameters are available by means of the Params component and may be edited. If read-out of variables fails, then zero value will be returned (False).
Available since version 2.5.

InsertToProject (_file_path : string, _ignore_warnings : bool, _only_geometry : bool) : bool

Function inserts definition of the structure described in the indicated STR file to the current project. If the parameters of the inserted structure have been read out and modified earlier, then their new values will be considered during structure insertion.
Available since version 2.5.

Robot Kernel

Available since version 2.5.

IRobotKernel

Main interface of the Robot program kernel.
Available since version 2.5.

Properties:

Version : double
version number
Available since version 2.5.

Preferences : IRobotKernelPreferences
settings for the Robot program kernel

Available since version 2.5.

ProjectPreferences : IRobotProjectPreferences
current project settings
Available since version 2.5.

Structure : IRobotStructure
interface describing the entire structure
Available since version 2.5.

CmpntFactory : IRobotComponentFactory

Available since version 2.5.

CalcEngine : IRobotCalcEngine

Available since version 2.5.

ProjectUniqueId : string
unique project identifier
Available since version 2.5.

ProgramVersion : string
designation of the program version
Available since version 5.5.

ProgramName : string
program name
Available since version 5.5.

ConcrReinfEngine : RConcrCalcEngine
calculation module for the concrete reinforcement
Available since version 5.5.

Operations:

ProjectNew (_prj_type : IRobotProjectType) : bool
Function creates a new project of the indicated type.
Available since version 2.5.

ProjectOpen (_file_path : string) : bool
Function opens the project saved in the indicated RTD format file.
Available since version 2.5.

ProjectSave ()
Function saves the current project to an RTD format file.
Available since version 2.5.

ProjectSaveAs (_file_path : string)
Function saves the current project to the indicated RTD format file.
Available since version 2.5.

GetExtension (_extmdl_name : string) : IDispatch
Function returns the interface for the module extending functionality of the Robot program kernel. If the extension with the indicated name has not been installed, then the function returns zero value (Nothing).
Available since version 2.5.

ProjectNewFromTemplate (_tmpl_file : string) : bool
Function creates a new document based on the template defined by the specified file.
Available since version 3.5.

IRobotKernelPreferences

Interface describing parameters of the Robot program kernel.
Available since version 2.5.

Operations:

GetLanguage (_lang_id : IRobotLanguage) : int
Function returns the current setting of the indicated language.
Available since version 2.5.

GetDirectory (_rob_dir : IRobotDirectory) : string
Function returns the full path to the selected folder.

Available since version 2.5.

SetLanguage (_lang_id : IRobotLanguage, _lang_num : int)

Available since version 2.5.

SetDirectoryExt (_dir_ext : IRobotDirectoryExtension, _full_path : string)

Function defines a folder extension. If an empty character string is given a as a full path to the extension, it will result in deleting the extension.

Meaning of the folder extension is best illustrated by the example.
If the SetDirectoryExt(I_DE_TEMPLATE, "C:\My Templates") function is activated, then from this moment on template files will be searched first in the C:\My Templates folder and next, in the standard folder with template files.

Available since version 3.5.

GetDirectoryExt (_dir_ext : IRobotDirectoryExtension) : string

Function returns folder extension. If for a specified folder no extension has been defined, function returns an empty character string.
Available since version 3.5.