**paceval_cRegisteredObject**

# paceval_cCleanupHandler
  * handle_CleanupHandler

# unsigned long registerPosition

+ paceval_cRegisteredObject
  (paceval_cCleanupHandler
  *handle_CleanupHandler_in)

+ ~paceval_cRegisteredObject()

+ void setRegisterPosition
  (unsigned long registerPosition_in)

+ unsigned long getRegister
  Position()

---

**paceval_cBaseAtomicGraphNode**

# bool * handle_stackHasCache
  OptionAvailable

# paceval_sNodeSpecificData
  * handle_sNodeSpecificData

+ paceval_cBaseAtomicGraph
  Node(paceval_cCleanupHandler
  *handle_CleanupHandler_in)

+ void initiateData(const
  char *operator_in, long
  valueNode1_in, long valueNode2
  _in, long resultNode_in, long
  position_in, const char *valueOperator_in)

+ ~paceval_cBaseAtomicGraph
  Node()

+ paceval_eListOfPointerTypes
  getPointerType()

+ long getPosition()

+ void getValueOperator
  (const paceval_eCalculation
  PrecisionTypes useCalculationPrecision
  _in, bool *valueOperatorIsTrusted
  _out, long double *valueAsLongDouble
  _out, double *valueAsDouble_out, float
  *valueAsFloat_out)

+ paceval_eOperatorTypes
  getOperator()

+ long * getValueField1()

+ long * getValueField2()

+ long * getResultField()

+ bool hasCacheOptionAvailable
  (unsigned long stackNumber_in)

+ bool CreateLookaheadCache
  Data()

+ void setLevelMultithread
  Jump(long levelMultithreadJump_in)

+ bool hasLevelMultithread
  Jump(long *levelMultithreadJump_out)

+ void setValueLevelMultithread
  (unsigned long idSingleCalculation
  _in, const paceval_eCalculationPrecision
  Types useCalculationPrecision_in, long
  double valueLevelMultithreadAsLongDouble
  _in, double valueLevelMultithreadAsDouble
  _in, float valueLevelMultithreadAsFloat_in,
  bool hasTrustedLevelMultithreadMinMaxResult
  _in, long double valueLevelMultithreadMinValue
  _in, long double valueLevelMultithreadMaxValue_in)

+ bool hasValueLevelMultithread
  (unsigned long idSingleCalculation
  _in, const paceval_eCalculationPrecision
  Types useCalculationPrecision_in, long
  double *valueLevelMultithreadAsLongDouble
  _out, double *valueLevelMultithreadAsDouble
  _out, float *valueLevelMultithreadAsFloat_out,
  bool *hasTrustedLevelMultithreadMinMaxResult
  _out, long double *valueLevelMultithreadMinValue
  _out, long double *valueLevelMultithreadMaxValue_out)

+ void setZeroCachingJump
  (long zeroCachingJump_in)

+ bool hasZeroCachingJump
  (long *zeroCachingJump_out)

---

**paceval_cAtomicGraphNode
AddValue**

+ paceval_cAtomicGraphNode
  AddValue(paceval_cCleanupHandler
  *handle_CleanupHandler_in)

+ void initializeDataAddVariable
  (const char *operator_in,
  long valueNode1_in, long position
  _in, long resultNode_in)

+ void initializeDataAddValue
  (const char *operator_in,
  long resultNode_in, long
  position_in, const char *valueOperator_in)