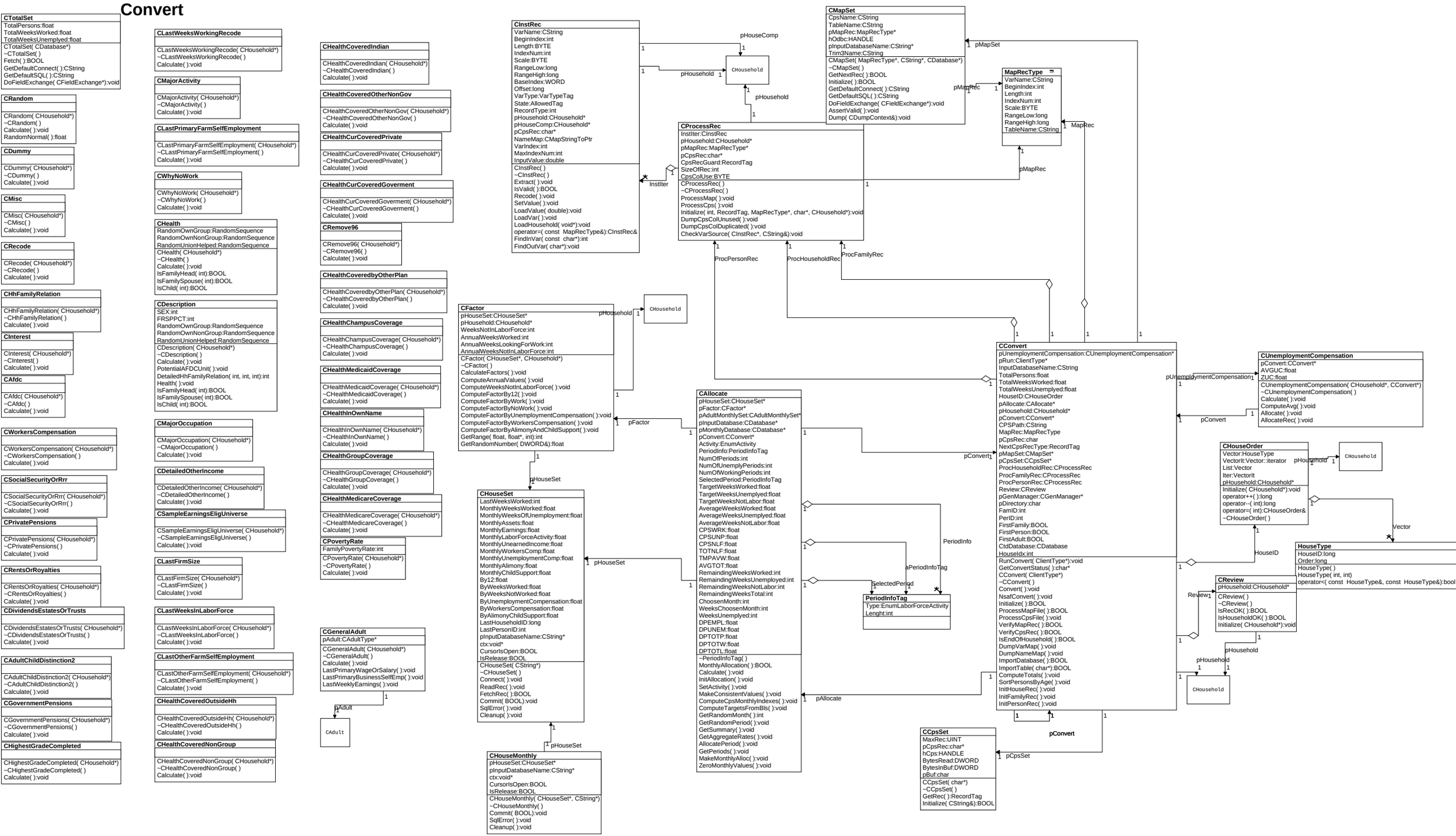
Convert


CTotalSet
TotalPersons:float
TotalWeeksWorked:float
TotalWeeksUnemplyed:float
CTotalSet(CDatabase*)
~CTotalSet( )
Fetch( ):BOOL
GetDefaultConnect( ):CString
GetDefaultSQL( ):CString
DoFieldExchange(CFieldExchange*):void
CRandom
CRandom(CHousehold*)
~CRandom( )
Calculate( ):void
RandomNormal( ):float
CDummy
CDummy(CHousehold*)
~CDummy( )
Calculate( ):void
CMisc
CMisc(CHousehold*)
~CMisc( )
Calculate( ):void
CRecode
CRecode(CHousehold*)
~CRecode( )
Calculate( ):void
CHhFamilyRelation
CHhFamilyRelation(CHousehold*)
~CHhFamilyRelation( )
Calculate( ):void
CInterest
CInterest(CHousehold*)
~CInterest( )
Calculate( ):void
CAfdc
CAfdc(CHousehold*)
~CAfdc( )
Calculate( ):void
CWorkersCompensation
CWorkersCompensation(CHousehold*)
~CWorkersCompensation( )
Calculate( ):void
CSocialSecurityOrRrr
CSocialSecurityOrRrr(CHousehold*)
~CSocialSecurityOrRrr( )
Calculate( ):void
CPrivatePensions
CPrivatePensions(CHousehold*)
~CPrivatePensions( )
Calculate( ):void
CRentsOrRoyalties
CRentsOrRoyalties(CHousehold*)
~CRentsOrRoyalties( )
Calculate( ):void
CDividendsEstatesOrTrusts
CDividendsEstatesOrTrusts(CHousehold*)
~CDividendsEstatesOrTrusts( )
Calculate( ):void
CAdultChildDistinction2
CAdultChildDistinction2(CHousehold*)
~CAdultChildDistinction2( )
Calculate( ):void
CGovernmentPensions
CGovernmentPensions(CHousehold*)
~CGovernmentPensions( )
Calculate( ):void
CHighestGradeCompleted
CHighestGradeCompleted(CHousehold*)
~CHighestGradeCompleted( )
Calculate( ):void
CLastWeeksWorkingRecode
CLastWeeksWorkingRecode(CHousehold*)
~CLastWeeksWorkingRecode( )
Calculate( ):void
CMajorActivity
CMajorActivity(CHousehold*)
~CMajorActivity( )
Calculate( ):void
CLastPrimaryFarmSelfEmployment
CLastPrimaryFarmSelfEmployment(CHousehold*)
~CLastPrimaryFarmSelfEmployment( )
Calculate( ):void
CWhyNoWork
CWhyNoWork(CHousehold*)
~CWhyNoWork( )
Calculate( ):void
CHealth
RandomOwnGroup:RandomSequence
RandomOwnNonGroup:RandomSequence
RandomUnionHelped:RandomSequence
CHealth(CHousehold*)
~CHealth( )
Calculate( ):void
IsFamilyHead(int):BOOL
IsFamilySpouse(int):BOOL
IsChild(int):BOOL
CDescription
SEX:int
FRSPPCT:int
RandomOwnGroup:RandomSequence
RandomOwnNonGroup:RandomSequence
RandomUnionHelped:RandomSequence
CDescription(CHousehold*)
~CDescription( )
Calculate( ):void
PotentialAFDCUnit( ):void
DetailedHhFamilyRelation(int, int, int):int
Health( ):void
IsFamilyHead(int):BOOL
IsFamilySpouse(int):BOOL
IsChild(int):BOOL
CMajorOccupation
CMajorOccupation(CHousehold*)
~CMajorOccupation( )
Calculate( ):void
CDetailedOtherIncome
CDetailedOtherIncome(CHousehold*)
~CDetailedOtherIncome( )
Calculate( ):void
CSampleEarningsEligUniverse
CSampleEarningsEligUniverse(CHousehold*)
~CSampleEarningsEligUniverse( )
Calculate( ):void
CLastFirmSize
CLastFirmSize(CHousehold*)
~CLastFirmSize( )
Calculate( ):void
CLastWeeksInLaborForce
CLastWeeksInLaborForce(CHousehold*)
~CLastWeeksInLaborForce( )
Calculate( ):void
CLastOtherFarmSelfEmployment
CLastOtherFarmSelfEmployment(CHousehold*)
~CLastOtherFarmSelfEmployment( )
Calculate( ):void
CHealthCoveredOutsideHh
CHealthCoveredOutsideHh(CHousehold*)
~CHealthCoveredOutsideHh( )
Calculate( ):void
CHealthCoveredNonGroup
CHealthCoveredNonGroup(CHousehold*)
~CHealthCoveredNonGroup( )
Calculate( ):void
CHealthCoveredIndian
CHealthCoveredIndian(CHousehold*)
~CHealthCoveredIndian( )
Calculate( ):void
CHealthCoveredOtherNonGov
CHealthCoveredOtherNonGov(CHousehold*)
~CHealthCoveredOtherNonGov( )
Calculate( ):void
CHealthCurCoveredPrivate
CHealthCurCoveredPrivate(CHousehold*)
~CHealthCurCoveredPrivate( )
Calculate( ):void
CHealthCurCoveredGoverment
CHealthCurCoveredGoverment(CHousehold*)
~CHealthCurCoveredGoverment( )
Calculate( ):void
CRemove96
CRemove96(CHousehold*)
~CRemove96( )
Calculate( ):void
CHealthCoveredbyOtherPlan
CHealthCoveredbyOtherPlan(CHousehold*)
~CHealthCoveredbyOtherPlan( )
Calculate( ):void
CHealthChampusCoverage
CHealthChampusCoverage(CHousehold*)
~CHealthChampusCoverage( )
Calculate( ):void
CHealthMedicaidCoverage
CHealthMedicaidCoverage(CHousehold*)
~CHealthMedicaidCoverage( )
Calculate( ):void
CHealthInOwnName
CHealthInOwnName(CHousehold*)
~CHealthInOwnName( )
Calculate( ):void
CHealthGroupCoverage
CHealthGroupCoverage(CHousehold*)
~CHealthGroupCoverage( )
Calculate( ):void
CHealthMedicareCoverage
CHealthMedicareCoverage(CHousehold*)
~CHealthMedicareCoverage( )
Calculate( ):void
CPovertyRate
FamilyPovertyRate:int
CPovertyRate(CHousehold*)
~CPovertyRate( )
Calculate( ):void
CGeneralAdult
pAdult:CAdultType*
CGeneralAdult(CHousehold*)
~CGeneralAdult( )
Calculate( ):void
LastPrimaryWageOrSalary( ):void
LastPrimaryBusinessSelfEmp( ):void
LastWeeklyEarnings( ):void
pAdult
CAdult
CInstRec
VarName:CString
BeginIndex:int
Length:BYTE
IndexNum:int
Scale:BYTE
RangeLow:long
RangeHigh:long
BaseIndex:WORD
Offset:long
VarType:VarTypeTag
State:AllowedTag
RecordType:int
pHousehold:CHousehold*
pHouseComp:CHousehold*
pCpsRec:char*
NameMap:CMapStringToPtr
VarIndex:int
MaxIndexNum:int
InputValue:double
CInstRec( )
~CInstRec( )
Extract( ):void
IsValid( ):BOOL
Recode( ):void
SetValue( ):void
LoadValue(double):void
LoadVar( ):void
LoadHousehold(void*):void
operator=(const MapRecType&):CInstRec&
FindInVar(const char*):int
FindOutVar(char*):void
pHouseComp
pHousehold
CHousehold
InstIter
CMapSet
CpsName:CString
TableName:CString
pMapRec:MapRecType*
hOdbc:HANDLE
pInputDatabaseName:CString*
Trim3Name:CString
CMapSet(MapRecType*, CString*, CDatabase*)
~CMapSet( )
GetNextRec( ):BOOL
Initialize( ):BOOL
GetDefaultConnect( ):CString
GetDefaultSQL( ):CString
DoFieldExchange(CFieldExchange*):void
AssertValid( ):void
Dump(CDumpContext&):void
pMapSet
pMapRec
MapRecType
VarName:CString
BeginIndex:int
Length:int
IndexNum:int
Scale:BYTE
RangeLow:long
RangeHigh:long
TableName:CString
MapRec
CProcessRec
InstIter:CInstRec
pHousehold:CHousehold*
pMapRec:MapRecType*
pCpsRec:char*
CpsRecGuard:RecordTag
SizeOfRec:int
CpsColUse:BYTE
CProcessRec( )
~CProcessRec( )
ProcessMap( ):void
ProcessCps( ):void
Initialize(int, RecordTag, MapRecType*, char*, CHousehold*):void
DumpCpsColUnused( ):void
DumpCpsColDuplicated( ):void
CheckVarSource(CInstRec*, CString&):void
ProcPersonRec
ProcHouseholdRec
ProcFamilyRec
CFactor
pHouseSet:CHouseSet*
pHousehold:CHousehold*
WeeksNotInLaborForce:int
AnnualWeeksWorked:int
AnnualWeeksLookingForWork:int
AnnualWeeksNotInLaborForce:int
CFactor(CHouseSet*, CHousehold*)
~CFactor( )
CalculateFactors( ):void
ComputeAnnualValues( ):void
ComputeWeeksNotInLaborForce( ):void
ComputeFactorBy12( ):void
ComputeFactorByWork( ):void
ComputeFactorByNoWork( ):void
ComputeFactorByUnemploymentCompensation( ):void
ComputeFactorByWorkersCompensation( ):void
ComputeFactorByAlimonyAndChildSupport( ):void
GetRange(float, float*, int):int
GetRandomNumber(DWORD&):float
pHousehold
CHousehold
pFactor
pHouseSet
CHouseSet
LastWeeksWorked:int
MonthlyWeeksWorked:float
MonthlyWeeksOfUnemployment:float
MonthlyAssets:float
MonthlyEarnings:float
MonthlyLaborForceActivity:float
MonthlyUnearnedIncome:float
MonthlyWorkersComp:float
MonthlyUnemploymentComp:float
MonthlyAlimony:float
MonthlyChildSupport:float
By12:float
ByWeeksWorked:float
ByWeeksNotWorked:float
ByUnemploymentCompensation:float
ByWorkersCompensation:float
ByAlimonyChildSupport:float
LastHouseholdID:long
LastPersonID:int
pInputDatabaseName:CString*
ctx:void*
CursorIsOpen:BOOL
IsRelease:BOOL
CHouseSet(CString*)
~CHouseSet( )
Connect( ):void
ReadRec( ):void
FetchRec( ):BOOL
Commit(BOOL):void
SqlError( ):void
Cleanup( ):void
pHouseSet
CHouseMonthly
pHouseSet:CHouseSet*
pInputDatabaseName:CString*
ctx:void*
CursorIsOpen:BOOL
IsRelease:BOOL
CHouseMonthly(CHouseSet*, CString*)
~CHouseMonthly( )
Commit(BOOL):void
SqlError( ):void
Cleanup( ):void
CAllocate
pHouseSet:CHouseSet*
pFactor:CFactor*
pAdultMonthlySet:CAdultMonthlySet*
pInputDatabase:CDatabase*
pMonthlyDatabase:CDatabase*
pConvert:CConvert*
Activity:EnumActivity
PeriodInfo:PeriodInfoTag
NumOfPeriods:int
NumOfUnemplyPeriods:int
NumOfWorkingPeriods:int
SelectedPeriod:PeriodInfoTag
TargetWeeksWorked:float
TargetWeeksUnemployed:float
TargetWeeksNotLabor:float
AverageWeeksWorked:float
AverageWeeksUnemployed:float
AverageWeeksNotLabor:float
CPSWRK:float
CPSUNP:float
CPSNLF:float
TOTNLF:float
TMPAVW:float
AVGTOT:float
RemaindingWeeksWorked:int
RemaindingWeeksUnemployed:int
RemaindingWeeksNotLabor:int
RemaindingWeeksTotal:int
ChoosenMonth:int
WeeksChoosenMonth:int
WeeksUnemplyed:int
DPEMPL:float
DPUNEM:float
DPTOTP:float
DPTOTW:float
DPTOTL:float
~PeriodInfoTag( )
MonthlyAllocation( ):BOOL
Calculate( ):void
InitAllocation( ):void
SetActivity( ):void
MakeConsistentValues( ):void
ComputeCpsMonthlyIndexes( ):void
ComputeTargetsFromBls( ):void
GetRandomMonth( ):int
GetRandomPeriod( ):void
GetSummary( ):void
GetAggregateRates( ):void
AllocatePeriod( ):void
GetPeriods( ):void
MakeMonthlyAlloc( ):void
ZeroMonthlyValues( ):void
pConvert
PeriodInfo
aPeriodInfoTag
SelectedPeriod
PeriodInfoTag
Type:EnumLaborForceActivity
Lenght:int
pAllocate
CConvert
pUnemploymentCompensation:CUnemploymentCompensation*
pRun:ClientType*
InputDatabaseName:CString
TotalPersons:float
TotalWeeksWorked:float
TotalWeeksUnemplyed:float
HouseID:CHouseOrder
pAllocate:CAllocate*
pHousehold:CHousehold*
pConvert:CConvert*
CPSPath:CString
MapRec:MapRecType
pCpsRec:char
NextCpsRecType:RecordTag
pMapSet:CMapSet*
pCpsSet:CCpsSet*
ProcHouseholdRec:CProcessRec
ProcFamilyRec:CProcessRec
ProcPersonRec:CProcessRec
Review:CReview
pGenManager:CGenManager*
pDirectory:char
FamID:int
PerID:int
FirstFamily:BOOL
FirstPerson:BOOL
FirstAdult:BOOL
CtdDatabase:CDatabase
HouseIdx:int
RunConvert(ClientType*):void
GetConvertStatus( ):char*
CConvert(ClientType*)
~CConvert( )
Convert( ):void
NsafConvert( ):void
Initialize( ):BOOL
ProcessMapFile( ):BOOL
ProcessCpsFile( ):void
VerifyMapRec( ):BOOL
VerifyCpsRec( ):BOOL
IsEndOfHousehold( ):BOOL
DumpVarMap( ):void
DumpNameMap( ):void
ImportDatabase( ):BOOL
ImportTable(char*):BOOL
ComputeTotals( ):void
SortPersonsByAge( ):void
InitHouseRec( ):void
InitFamilyRec( ):void
InitPersonRec( ):void
pUnemploymentCompensation
pConvert
CUnemploymentCompensation
pConvert:CConvert*
AVGUC:float
ZUC:float
CUnemploymentCompensation(CHousehold*, CConvert*)
~CUnemploymentCompensation( )
Calculate( ):void
ComputeAvg( ):void
Allocate( ):void
AllocateRec( ):void
CHouseOrder
Vector:HouseType
VectorIt:Vector::iterator
List:Vector
Iter:VectorIt
pHousehold:CHousehold*
Initialize(CHousehold*):void
operator++( ):long
operator--(int):long
operator=(int):CHouseOrder&
~CHouseOrder( )
pHousehold
CHousehold
Vector
HouseID
HouseType
HouseID:long
Order:long
HouseType( )
HouseType(int, int)
operator<(const HouseType&, const HouseType&):bool
Review
CReview
pHousehold:CHousehold*
CReview( )
~CReview( )
IsRecOK( ):BOOL
IsHouseholdOK( ):BOOL
Initialize(CHousehold*):void
pHousehold
CHousehold
CCpsSet
MaxRec:UINT
pCpsRec:char*
hCps:HANDLE
BytesRead:DWORD
BytesInBuf:DWORD
pBuf:char
CCpsSet(char*)
~CCpsSet( )
GetRec( ):RecordTag
Initialize(CString&):BOOL
pCpsSet