NAME

PDBFileUtil

SYNOPSIS

use PDBFileUtil ;

use PDBFileUtil qw(:all);

DESCRIPTION

PDBFileUtil module provides the following functions:

GenerateAtomOrHetatmRecordLine, GenerateAtomRecordLine, GenerateConectRecordLine, GenerateEndRecordLine, GenerateHeaderRecordLine, GenerateHeaderRecordTimeStamp, GenerateHetatmRecordLine, GenerateTerRecordLine, GetAllResidues, GetChainsAndResidues, GetConectRecordLines, GetExperimentalTechnique, GetExperimentalTechniqueResolution, GetMinMaxCoords, GetPDBRecordType, GetRecordTypesCount, IsAtomRecordType, IsConectRecordType, IsEndmdlRecordType, IsHeaderRecordType, IsHetatmRecordType, IsMasterRecordType, IsModelRecordType, IsPDBFile, IsSeqresRecordType, IsTerRecordType, ParseAtomOrHetatmRecordLine, ParseAtomRecordLine, ParseConectRecordLine, ParseExpdtaRecordLine, ParseHeaderRecordLine, ParseHetatmRecordLine, ParseMasterRecordLine, ParseRemark2ResolutionRecordLine, ParseSeqresRecordLine, ParseTerRecordLine, ReadPDBFile

METHODS

GenerateAtomOrHetatmRecordLine

```
$RecordLine = GenerateAtomOrHetatmRecordLine($RecordType,
  $AtomNumber, $AtomName, $AlternateLocation, $ResidueName,
  $ChainID, $ResidueNumber, $InsertionCode, $X, $Y, $Z,
  $Occupancy, $TemperatureFactor, $SegmentID,
  $ElementSymbol, $AtomCharge);
```

Returns ATOM or HETATM record line.

GenerateAtomRecordLine

```
$RecordLine = GenerateAtomRecordLine($AtomNumber,
  $AtomName, $AlternateLocation, $ResidueName, $ChainID,
  $ResidueNumber, $InsertionCode, $X, $Y, $Z, $Occupancy,
  $TemperatureFactor, $SegmentID, $ElementSymbol, $AtomCharge);
```

Returns ATOM record line.

GenerateConectRecordLine

```
$RecordLine = GenerateConectRecordLine($AtomNum, $BondedAtomNum1,
  $BondedAtomNum2, $BondedAtomNum3, $BondedAtomNum4,
  $HBondedAtomNum1, $HBondedAtomNum2, $SaltBridgedAtomNum1,
  $HBondedAtomNum3, $HBondedAtomNum4, $SaltBridgedAtomNum2);
```

Returns CONECT record line.

GenerateHeaderRecordLine

```
$RecordLine = GenerateHeaderRecordLine($IDCode, [$Classification,
  $Date]);
```

Returns HEADER record line.

GenerateHeaderRecordTimeStamp

```
$Date = GenerateHeaderRecordTimeStamp();
```

Returns PDB header time stamp.

GenerateHetatmRecordLine

```
$RecordLine = GenerateHetatmRecordLine($AtomNumber, $AtomName,
```

```
    $AlternateLocation, $ResidueName, $ChainID, $ResidueNumber,
    $InsertionCode, $X, $Y, $Z, $Occupancy, $TemperatureFactor,
    $SegmentID, $ElementSymbol, $AtomCharge);
```

Returns HETATM record line.

GenerateEndRecordLine

```
    $RecordLine = GenerateEndRecordLine();
```

Returns END record line.

GenerateTerRecordLine

```
    $RecordLine = GenerateTerRecordLine($SerialNumber, [$ResidueName,
      $ChainID, $ResidueNumber, $InsertionCode]);
```

Returns TER record line.

GetAllResidues

```
    $ResiduesDataRef = GetAllResidues($PDBRecordLinesRef);
```

Gets residue information using ATOM/HETATM records and returns a reference to a hash with following key/value pairs:

```
    $ResiduesDataRef->{ResidueNames} - Array of all the residues
    $ResiduesDataRef->{ResidueCount}{$ResidueName} - Count of residues
    $ResiduesDataRef->{AtomResidueNames}} - Array of all ATOM residues
    $ResiduesDataRef->{AtomResidueCount}{$ResidueName} - Count of
       residues in ATOM records
    $ResiduesDataRef->{HetatomResidueNames} - List of all HETATM
       residues
    $ResiduesDataRef->{HetatmResidueCount}{$ResidueName} - Count of
      residues HETATM records
```

ATOM/HETATM records after the first ENDMDL records are simply ingnored.

GetChainsAndResidues

```
    $ChainsDataRef = GetChainsAndResidues($PDBRecordLinesRef,
      [$RecordsSource, $GetChainResiduesBeyondTERFlag,
      $GetRecordLinesFlag]);
```

Gets chains and residue information using ATOM/HETATM or SEQRES records and returns a reference to a hash with these keys:

```
    $ChainsDataRef->{ChainIDs} - List of chain IDs with 'None' for
      no IDs
    $ChainsDataRef->{Residues}{$ChainID} - List of residues in order
      of their appearance in a chain
    $ChainsDataRef->{ResidueCount}{$ChainID}{$ResidueName} - Count of
      residues in a chain
```

Chains and residue data can be extacted using either ATOM/HETATM records or SEQRES records. ATOM/HETATM records after the first ENDMDL records are simply ingnored.

GetConectRecordLines

```
    $ConectRecordLinesRef = GetConectRecordLines($PDBRecordLinesRef,
      $AtomNumbersMapRef);
```

Collects CONECT record lines for specific atom number, modified specified data to exclude any atom number not present in the list of specified atom numbers and returns a reference to list of CONECT record lines.

GetExperimentalTechnique

```
    $ExperimentalTechnique = GetExperimentalTechnique($PDBRecordLinesRef);
```

Returns *ExperimentalTechnique* value retrieved from EXPDATA record line.

GetExperimentalTechniqueResolution

```
($Resolution, $ResolutionUnits) = GetExperimentalTechniqueResolution(
                                  $PDBRecordLinesRef);
```

Returns *Resolution* and *ResolutionUnits* values from REMARK 2 RESOLUTION record line.

GetMinMaxCoords

```
($XMin, $YMin, $ZMin, $XMax, $YMax, $ZMax) =
  GetMinMaxCoords($PDBRecordLinesRef);
```

Returns minimum and maximum XYZ coordinates for ATOM/HETATM records.

GetPDBRecordType

```
$RecordType = GetPDBRecordType($RecordLine);
```

Returns type of *RecordLine*.

GetRecordTypesCount

```
$RecordTypeDataRef = GetRecordTypesCount($PDBRecordLinesRef,
  [$SpecifiedRecordType, $GetRecordLinesFlag]);
```

Counts the number of each record type or a $SpecifiedRecordType and returns a reference to data type with following key/value pairs:

```
$RecordTypeDataRef->{RecordTypes} - An array of unique record types
   in order of their presence in the file
$RecordTypeDataRef->{Count}{$RecordType} - Count of each record type
$RecordTypeDataRef->{Lines}{$RecordType} - Optional lines data for a
  specific record type.
```

IsAtomRecordType

```
$Status = IsAtomRecordType($RecordLine);
```

Returns 1 or 0 based on whether it's a ATOM record line.

IsConectRecordType

```
$Status = IsAtomConectType($RecordLine);
```

Returns 1 or 0 based on whether it's a CONECT record line.

IsEndmdlRecordType

```
$Status = IsEndmdlRecordType($RecordLine);
```

Returns 1 or 0 based on whether it's a ENDMDL a record line.

IsHeaderRecordType

```
$Status = IsHeaderRecordType($RecordLine);
```

Returns 1 or 0 based on whether it's a HEADER a record line.

IsHetatmRecordType

```
$Status = IsHetatmRecordType($RecordLine);
```

Returns 1 or 0 based on whether it's a HETATM a record line.

IsMasterRecordType

```
$Status = IsMasterRecordType($RecordLine);
```

Returns 1 or 0 based on whether it's a MASTER a record line.

IsModelRecordType

```
$Status = IsModelRecordType($RecordLine);
```

Returns 1 or 0 based on whether it's a MODEL record line.

IsPDBFile

```
$Status = IsPDBFile($PDBFile);
```

Returns 1 or 0 based on whether it's a PDB file.

IsSeqresRecordType

```
$Status = IsSeqresRecordType($RecordLine);
```

Returns 1 or 0 based on whether it's SEQRES a record line.

IsTerRecordType

```
$Status = IsTerRecordType($RecordLine);
```

Returns 1 or 0 based on whether it's a TER record line.

ParseAtomOrHetatmRecordLine

```
($AtomNumber, $AtomName, $AlternateLocation, $ResidueName, $ChainID,
  $ResidueNumber, $InsertionCode, $X, $Y, $Z, $Occupancy,
  $TemperatureFactor, $SegmentID, $ElementSymbol, $AtomCharge) =
  ParseAtomOrHetatmRecordLine($RecordLine);
```

Parses ATOM or HETATM record line.

ParseAtomRecordLine

```
($AtomNumber, $AtomName, $AlternateLocation, $ResidueName, $ChainID,
  $ResidueNumber, $InsertionCode, $X, $Y, $Z, $Occupancy,
  $TemperatureFactor, $SegmentID, $ElementSymbol, $AtomCharge) =
  ParseAtomRecordLine($RecordLine);
```

Parses ATOM record line.

ParseConectRecordLine

```
($AtomNum, $BondedAtomNum1, $BondedAtomNum2, $BondedAtomNum3,
   $BondedAtomNum4, $HBondedAtomNum1, $HBondedAtomNum2,
   $SaltBridgedAtomNum1, $HBondedAtomNum3, $HBondedAtomNum4,
   $SaltBridgedAtomNum2) = ParseConectRecordLine($RecordLine);
```

Parses CONECT record line.

ParseExpdtaRecordLine

```
($ContinuationNum, $ExperimentalTechnique) = ParseExpdtaRecordLine($Line);
```

Parses EXPDTA record line.

ParseHeaderRecordLine

```
($Classification, $DepositionDate, $IDCode) = ParseHeaderRecordLine($RecordLine);
```

Parses HEADER record line

ParseHetatmRecordLine

```
($AtomNumber, $AtomName, $AlternateLocation, $ResidueName, $ChainID,
  $ResidueNumber, $InsertionCode, $X, $Y, $Z, $Occupancy,
  $TemperatureFactor, $SegmentID, $ElementSymbol, $AtomCharge) =
  ParseHetatmRecordLine($RecordLine);
```

Parses HETATM record line.

ParseMasterRecordLine

```
($NumOfRemarkRecords, $NumOfHetRecords, $NumOfHelixRecords,
  $NumOfSheetRecords, $NumOfTurnRecords, $NumOfSiteRecords,
  $NumOfTransformationsRecords, $NumOfAtomAndHetatmRecords,
  $NumOfTerRecords, $NumOfConectRecords, $NumOfSeqresRecords) =
  ParseMasterRecordLine($RecordLine);
```

Parses MASTER ecord line.

ParseRemark2ResolutionRecordLine

```
($Resolution, $ResolutionUnits) = ParseRemark2ResolutionRecordLine(
                                  $RecordLine);
```

Parses REMARK 2 RESOLUTION record line.

ParseSeqresRecordLine

```
($RecordSerialNumber, $ChainID, $NumOfResidues, $ResidueNames) =
  ParseSeqresRecordLine($RecordLine);
```

Parses SEQRES record line.

ParseTerRecordLine

```
($SerialNumber, $ResidueName, $ChainID, $ResidueNumber, $InsertionCode) =
  ParseTerRecordLine($RecordLine);
```

Parses TER record line.

ReadPDBFile

```
$PDBRecordLinesRef = ReadPDBFile($PDBFile);
```

Reads PDB file and returns reference to record lines.

AUTHOR

Manish Sud <msud@san.rr.com>

SEE ALSO

FileUtil.pm, SequenceFileUtil.pm, TextUtil.pm

COPYRIGHT